Moussaif Hermes-Fugu Fusion HyperSkills KB v2.0

Agentic AI Engineering toward AGI-Level Multi-Model Orchestration

نسخة موسعة مدمجة من قاعدة HyperSkills: Fusion Hermes-Fugu لا تمحو الأصل، بل ترفعه إلى هندسة AGI متعددة النماذج، متعددة المحاكم، متعددة الأوعية.

تضيف هذه النسخة طبقة v2 داخل الجسد المعرفي نفسه: Fugu-style routing, MCD hyperloops, hyperskills, courts, demolition governance, GraphRAG abstraction, provider OpenRouter/NVIDIA orchestration, schemata. implementation and program, evaluation

الغرض: تحويل تعدد النماذج من تجميع سطحي إلى نظام تشغيل معرفي يختار، يوزع، يهاجم، يصهر، يتعلم، ويعيد بناء ذاكرته.

```
Input -> Difference -> MCD -> Vascular Router -> Multi-Agent Paths -> Court -> Tools/RAG/GraphRAG -> Fusion -> World Eff
```

إنجاز وتركيب: Moussaif Zouhair / زهير مساعف

contract preservation and Identity - وحدة 002

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

عقد حفظ الأصل ورفعه

الأصل لا يمحى: v1 spine يبقى باعتباره قاعدة Fusion Hermes-Fugu التي عرّفت abstraction, provider court, model fusion, routing, hyperskill and primitives. النسخة v2 لا تكون جزءا ثانيا؛ إنها إعادة حقن للكتاب نفسه: كل مفهوم سابق يحصل على تعريف أدق، حد معرفي، آلية تشغيل، أو أداة قياس.

مرتبة	وظيفة	قاعدة الاستعمال
ثابت/مصدر	معلومة من وثيقة أو ورقة	لا تعمم خارج المصدر
تركيب نظري	جمع بين حقول	يحتاج اختبارا
أداة	بروتوكول تشغيل	تقاس بنتيجة
فرضية	دعوى مستقبلية	تدخل في برنامج بحث

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 003 - contract preservation and Identity

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

هوية الكتاب الأصلي

عنوان الأصل: Moussaif Hermes-Fugu Fusion HyperSkills KB. أطروحته المركزية: تحويل تعدد النماذج من قائمة tools إلى نظام حاكم يختار، يوزع، يتحقق، ويصهر النتائج. السؤال الأكبر: كيف يصبح Hermes قادرا على استخراج أعلى قدرة من نماذج مفتوحة أو مجانية أو قابلة للاستضافة دون الوقوع في فوضى تعدد الوكلاء؟

مرتبة	وظيفة	قاعدة الاستعمال
ثابت/مصدر	معلومة من وثيقة أو ورقة	لا تعمم خارج المصدر
تركيب نظري	جمع بين حقول	يحتاج اختبارا
أداة	بروتوكول تشغيل	تقاس بنتيجة
فرضية	دعوى مستقبلية	تدخل في برنامج بحث

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 004 - contract preservation and Identity

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

سلم مراتب المعرفة

كل ادعاء في هذا الكتاب يثبت في رتبة: ثابت بمصدر، راجح هندسيا، تركيب نظري، فرضية قابلة للاختبار، استعارة منتجة، أو target. design مصطلحات compute MCAI/MCD/vascular تعرض كعدسات توليدية وهندسية لا كحقائق علمية مثبتة بذاتها.

مرتبة	وظيفة	قاعدة الاستعمال
ثابت/مصدر	معلومة من وثيقة أو ورقة	لا تعمم خارج المصدر
تركيب نظري	جمع بين حقول	يحتاج اختبارا
أداة	بروتوكول تشغيل	تقاس بنتيجة
فرضية	دعوى مستقبلية	تدخل في برنامج بحث

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 005 - contract preservation and Identity

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

مصفوفة النقص إلى البناء

نقص v1 التعريفي عولج بقاموس حدودي. نقص v1 القياسي عولج بمؤشرات MCD, FEQ, RCR, SRI, WEI. نقص v1 التطبيقي عولج بملفات router/court/fusion/memory/provider وبسياسة trace إلزامية.

مرتبة	وظيفة	قاعدة الاستعمال
ثابت/مصدر	معلومة من وثيقة أو ورقة	لا تعمم خارج المصدر
تركيب نظري	جمع بين حقول	يحتاج اختبارا
أداة	بروتوكول تشغيل	تقاس بنتيجة
فرضية	دعوى مستقبلية	تدخل في برنامج بحث

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

ranks evidence method, Index, - وحدة 006

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
1-1	Cover	حقن داخلي / توسعة تأسيسية
2-5	contract preservation and Identity	حقن داخلي / توسعة تأسيسية
6-20	ranks evidence method, Index,	حقن داخلي / توسعة تأسيسية
21-80	definitions technical Core	حقن داخلي / توسعة تأسيسية
81-150	ontology hyperskill AGI	حقن داخلي / توسعة تأسيسية
151-220	patterns and laws Hyperloop	حقن داخلي / توسعة تأسيسية

- حفظ spine v1 دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

ranks evidence method, Index, - وحدة 007

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
2-5	contract preservation and Identity	حقن داخلي / توسعة تأسيسية
6-20	ranks evidence method, Index,	حقن داخلي / توسعة تأسيسية
21-80	definitions technical Core	حقن داخلي / توسعة تأسيسية
81-150	ontology hyperskill AGI	حقن داخلي / توسعة تأسيسية
151-220	patterns and laws Hyperloop	حقن داخلي / توسعة تأسيسية
221-290	router fusion Hermes-Fugu	حقن داخلي / توسعة تأسيسية

- حفظ v1 spine دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

ranks evidence method, Index, - وحدة 008

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
6-20	ranks evidence method, Index,	حقن داخلي / توسعة تأسيسية
21-80	definitions technical Core	حقن داخلي / توسعة تأسيسية
81-150	ontology hyperskill AGI	حقن داخلي / توسعة تأسيسية
151-220	patterns and laws Hyperloop	حقن داخلي / توسعة تأسيسية
221-290	router fusion Hermes-Fugu	حقن داخلي / توسعة تأسيسية
291-350	layer provider OpenRouter/NVIDIA	حقن داخلي / توسعة تأسيسية

- حفظ spine v1 دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

ranks evidence method, Index, - وحدة 009

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
21-80	definitions technical Core	حقن داخلي / توسعة تأسيسية
81-150	ontology hyperskill AGI	حقن داخلي / توسعة تأسيسية
151-220	patterns and laws Hyperloop	حقن داخلي / توسعة تأسيسية
221-290	router fusion Hermes-Fugu	حقن داخلي / توسعة تأسيسية
291-350	layer provider OpenRouter/NVIDIA	حقن داخلي / توسعة تأسيسية
351-410	engineering AGI and theories Moussaif	حقن داخلي / توسعة تأسيسية

- حفظ spine v1 دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

ranks evidence method, Index, - وحدة 010

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
81-150	ontology hyperskill AGI	حقن داخلي / توسعة تأسيسية
151-220	patterns and laws Hyperloop	حقن داخلي / توسعة تأسيسية
221-290	router fusion Hermes-Fugu	حقن داخلي / توسعة تأسيسية
291-350	layer provider OpenRouter/NVIDIA	حقن داخلي / توسعة تأسيسية
351-410	engineering AGI and theories Moussaif	حقن داخلي / توسعة تأسيسية
411-470	protocols metrics, Tools,	حقن داخلي / توسعة تأسيسية

- حفظ v1 spine دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

ranks evidence method, Index, - وحدة 011

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
151-220	patterns and laws Hyperloop	حقن داخلي / توسعة تأسيسية
221-290	router fusion Hermes-Fugu	حقن داخلي / توسعة تأسيسية
291-350	layer provider OpenRouter/NVIDIA	حقن داخلي / توسعة تأسيسية
351-410	engineering AGI and theories Moussaif	حقن داخلي / توسعة تأسيسية
411-470	protocols metrics, Tools,	حقن داخلي / توسعة تأسيسية
471-520	design benchmark and Evaluation	حقن داخلي / توسعة تأسيسية

- حفظ spine v1 دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

ranks evidence method, Index, - وحدة 012

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
221-290	router fusion Hermes-Fugu	حقن داخلي / توسعة تأسيسية
291-350	layer provider OpenRouter/NVIDIA	حقن داخلي / توسعة تأسيسية
351-410	engineering AGI and theories Moussaif	حقن داخلي / توسعة تأسيسية
411-470	protocols metrics, Tools,	حقن داخلي / توسعة تأسيسية
471-520	design benchmark and Evaluation	حقن داخلي / توسعة تأسيسية
521-570	courts demolition and program Research	حقن داخلي / توسعة تأسيسية

- حفظ spine v1 دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

ranks evidence method, Index, - وحدة 013

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
291-350	layer provider OpenRouter/NVIDIA	حقن داخلي / توسعة تأسيسية
351-410	engineering AGI and theories Moussaif	حقن داخلي / توسعة تأسيسية
411-470	protocols metrics, Tools,	حقن داخلي / توسعة تأسيسية
471-520	design benchmark and Evaluation	حقن داخلي / توسعة تأسيسية
521-570	courts demolition and program Research	حقن داخلي / توسعة تأسيسية
571-610	blueprints Implementation	حقن داخلي / توسعة تأسيسية

- حفظ v1 spine دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

ranks evidence method, Index, - وحدة 014

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
351-410	engineering AGI and theories Moussaif	حقن داخلي / توسعة تأسيسية
411-470	protocols metrics, Tools,	حقن داخلي / توسعة تأسيسية
471-520	design benchmark and Evaluation	حقن داخلي / توسعة تأسيسية
521-570	courts demolition and program Research	حقن داخلي / توسعة تأسيسية
571-610	blueprints Implementation	حقن داخلي / توسعة تأسيسية
611-620	log build glossary, Sources,	حقن داخلي / توسعة تأسيسية

- حفظ v1 spine دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Memory of Reasons.
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

ranks evidence method, Index, - وحدة 015

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
411-470	protocols metrics, Tools,	حقن داخلي / توسعة تأسيسية
471-520	design benchmark and Evaluation	حقن داخلي / توسعة تأسيسية
521-570	courts demolition and program Research	حقن داخلي / توسعة تأسيسية
571-610	blueprints Implementation	حقن داخلي / توسعة تأسيسية
611-620	log build glossary, Sources,	حقن داخلي / توسعة تأسيسية
1-1	Cover	حقن داخلي / توسعة تأسيسية

- حفظ v1 spine دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

ranks evidence method, Index, - وحدة 016

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
471-520	design benchmark and Evaluation	حقن داخلي / توسعة تأسيسية
521-570	courts demolition and program Research	حقن داخلي / توسعة تأسيسية
571-610	blueprints Implementation	حقن داخلي / توسعة تأسيسية
611-620	log build glossary, Sources,	حقن داخلي / توسعة تأسيسية
1-1	Cover	حقن داخلي / توسعة تأسيسية
2-5	contract preservation and Identity	حقن داخلي / توسعة تأسيسية

- حفظ spine v1 دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

ranks evidence method, Index, - وحدة 017

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
521-570	courts demolition and program Research	حقن داخلي / توسعة تأسيسية
571-610	blueprints Implementation	حقن داخلي / توسعة تأسيسية
611-620	log build glossary, Sources,	حقن داخلي / توسعة تأسيسية
1-1	Cover	حقن داخلي / توسعة تأسيسية
2-5	contract preservation and Identity	حقن داخلي / توسعة تأسيسية
6-20	ranks evidence method, Index,	حقن داخلي / توسعة تأسيسية

- حفظ v1 spine دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

ranks evidence method, Index, - وحدة 018

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
571-610	blueprints Implementation	حقن داخلي / توسعة تأسيسية
611-620	log build glossary, Sources,	حقن داخلي / توسعة تأسيسية
1-1	Cover	حقن داخلي / توسعة تأسيسية
2-5	contract preservation and Identity	حقن داخلي / توسعة تأسيسية
6-20	ranks evidence method, Index,	حقن داخلي / توسعة تأسيسية
21-80	definitions technical Core	حقن داخلي / توسعة تأسيسية

- حفظ v1 spine دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

ranks evidence method, Index, - وحدة 019

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
611-620	log build glossary, Sources,	حقن داخلي / توسعة تأسيسية
1-1	Cover	حقن داخلي / توسعة تأسيسية
2-5	contract preservation and Identity	حقن داخلي / توسعة تأسيسية
6-20	ranks evidence method, Index,	حقن داخلي / توسعة تأسيسية
21-80	definitions technical Core	حقن داخلي / توسعة تأسيسية
81-150	ontology hyperskill AGI	حقن داخلي / توسعة تأسيسية

- حفظ spine v1 دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

ranks evidence method, Index, - وحدة 020

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الفهرس هنا خريطة تشغيل لا زينة. كل باب يبني طبقة في نظام Hermes: من التعريف إلى المهارة، من الحلقة إلى noitartsehcro، من المحكمة إلى الأثر العالمي. لا توجد إضافات في آخر الكتاب كملحق منفصل؛ كل توسعة مرتبطة بنقص أو بذرة في v1.

الصفحات	الباب	نوع الحقن
1-1	Cover	حقن داخلي / توسعة تأسيسية
2-5	contract preservation and Identity	حقن داخلي / توسعة تأسيسية
6-20	ranks evidence method, Index,	حقن داخلي / توسعة تأسيسية
21-80	definitions technical Core	حقن داخلي / توسعة تأسيسية
81-150	ontology hyperskill AGI	حقن داخلي / توسعة تأسيسية
151-220	patterns and laws Hyperloop	حقن داخلي / توسعة تأسيسية

- حفظ spine v1 دون إلغائه.
- إعادة تعريف المفاهيم الواسعة.
- تحويل loops إلى hyperskills قابلة للقياس.
- إدخال محاكم هدم قبل النموذج النهائي.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

definitions technical Core - وحدة 021

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperskill = مهارة فائقة قابلة للاستدعاء لها trigger, state, role, tools, verifier memory ,update and rule. retirement

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv: 2310.11511 ،reflection through critique generation, retrieval, adaptive tokens. - Self-RAG

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

definitions technical Core - وحدة 022

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperloop = حلقة تشغيلية تعود إلى نفسها بتعديل في الذاكرة أو السياسة أو الأدلة، ليست تكرارا أعمى.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: split. speed-quality Fugu/Fugu-Ultra scaffolds, adaptive orchestration, multi-agent learned 2606.21228: arXiv - Report Technical Fugu Sakana

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

definitions technical Core - وحدة 023

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: MCD = مقياس فرق مركب يحدد متى يحتاج الطلب إلى routing بسيط أو court fusion كامل.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: synthesis. verification, delegation, selection, model for interface API single sakana.ai/fugu-release: - Release Fugu Sakana

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

definitions technical Core - وحدة 024

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Compute Vascular = توزيع الحساب لا حسب القوة العامة للنموذج بل حسب الوعاء: retrieve, verify, code, plan, synthesize.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: OpenRouter Models Free Collection - :https://openrouter.ai/collections/free-models, free routes, provider/privacy router, model warnings, catalog.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

definitions technical Core - وحدة 025

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Orchestration Fugu-style = نظام يقرر query per كيف يستدعي agents of team ثم يصهرها في واجهة واحدة.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: NVIDIA NIM LLM API - docs.nvidia.com/nim/large-language-models/latest/api-reference.html: OpenAI-compatible inference NIM deployable for services.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

definitions technical Core - وحدة 026

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Court Fusion = محكمة تقارن outputs وتفصل بين action، final and risk, mechanism, evidence, claim

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - ReAct: 2210.03629، with acting and reasoning interleaving tools/environments

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

definitions technical Core - وحدة 027

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Memory Self-Reconstruction = ذاكرة تعيد بناء routing policy بناء على سبب الفشل أو النجاح لا على النص النهائي.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Thoughts of Tree: 2305.10601، reasoning multiple over search، and self-evaluation paths، backtracking.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback، rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement، condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

definitions technical Core - وحدة 028

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperskill = مهارة فائقة قابلة للاستدعاء لها trigger, state, role, tools, memory verifier, and update, rule. retirement

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Self-Consistency: 2203.11171. multiple sampling then paths reasoning answers. consistent selecting

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

definitions technical Core - وحدة 029

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperloop = حلقة تشغيلية تعود إلى نفسها بتعديل في الذاكرة أو السياسة أو الأدلة، ليست تكرارا أعمى.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Reflexion: 2303.11366. episodic and feedback verbal for memory agents.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

definitions technical Core - وحدة 030

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: MCD = مقياس فرق مركب يحدد متى يحتاج الطلب إلى routing بسيط أو court fusion كامل.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - RAG :2005.11401 .knowledge-intensive for memory non-parametric plus parametric generation

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

definitions technical Core - وحدة 031

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Compute Vascular = توزيع الحساب لا حسب القوة العامة للنموذج بل حسب الوعاء: retrieve, verify, code, plan, synthesize.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: GraphRAG - Microsoft + arXiv: surveys, graph-structured retrieval, multi-hop communities, relations.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

definitions technical Core - وحدة 032

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Orchestration Fugu-style = نظام يقرر query per كيف يستدعي agents of team ثم يصهرها في واجهة واحدة.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Self-RAG: 2310.11511. reflection through critique generation, retrieval, adaptive tokens.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 033 - definitions technical Core

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Court Fusion = محكمة تقارن outputs وتفصل بين action، final and risk, mechanism, evidence, claim

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: Technical Fugu Sakana - arXiv: 2606.21228 multi-agent learned adaptive orchestration, scaffolds Fugu/Fugu-Ultra speed-quality split.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 034 - definitions technical Core

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Memory Self-Reconstruction = ذاكرة تعيد بناء routing policy بناء على سبب الفشل أو النجاح لا على النص النهائي.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: Release Fugu Sakana - sakana.ai/fugu-release/: synthesis، verification, delegation, selection, model for interface API single

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 035 - definitions technical Core

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperskill = مهارة فائقة قابلة للاستدعاء لها trigger, state, role, tools, memory verifier, and update, rule. retirement

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: Collection Models Free OpenRouter - :https://openrouter.ai/collections/free-models, free routes, provider/privacy router, model warnings, catalog.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

definitions technical Core - وحدة 036

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperloop = حلقة تشغيلية تعود إلى نفسها بتعديل في الذاكرة أو السياسة أو الأدلة، ليست تكرارا أعمى.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: API LLM NIM NVIDIA - : NIM deployable for interface inference OpenAI-compatible docs.nvidia.com/nim/large-language-models/latest/api-reference.html services.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 037 - definitions technical Core

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: MCD = مقياس فرق مركب يحدد متى يحتاج الطلب إلى routing بسيط أو court fusion كامل.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - ReAct: 2210.03629. with acting and reasoning interleaving tools/environments

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

definitions technical Core - وحدة 038

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Compute Vascular = توزيع الحساب لا حسب القوة العامة للنموذج بل حسب الوعاء: retrieve, verify, code, plan, synthesize.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Thoughts of Tree 2305.10601: search over multiple reasoning, backtracking, and self-evaluation paths.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

definitions technical Core - وحدة 039

expansion: integrated 2 Version حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Orchestration Fugu-style = نظام يقرر query per كيف يستدعي agents of team ثم يصهرها في واجهة واحدة.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Self-Consistency :2203.11171 sampling multiple reasoning paths then consistent selecting answers.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

definitions technical Core - وحدة 040

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Court Fusion = محكمة تقارن outputs وتفصل بين action، final and risk, mechanism, evidence, claim

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Reflexion: 2303.11366 verbal feedback and episodic memory for agents.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 041 - definitions technical Core

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Memory Self-Reconstruction = ذاكرة تعيد بناء routing policy بناء على سبب الفشل أو النجاح لا على النص النهائي.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - RAG: 2005.11401، knowledge-intensive for memory non-parametric plus parametric generation.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

definitions technical Core - وحدة 042

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperskill = مهارة فائقة قابلة للاستدعاء لها trigger, state, role, tools, memory verifier, and update, rule. retirement

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: GraphRAG - Microsoft + arXiv: surveys, graph-structured retrieval, multi-hop communities. relations

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

definitions technical Core - وحدة 043

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperloop = حلقة تشغيلية تعود إلى نفسها بتعديل في الذاكرة أو السياسة أو الأدلة، ليست تكرارا أعمى.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Self-RAG: 2310.11511. reflection through critique generation, retrieval, adaptive tokens.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

definitions technical Core - وحدة 044

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: MCD = مقياس فرق مركب يحدد متى يحتاج الطلب إلى routing بسيط أو court fusion كامل.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: split. speed-quality Fugu/Fugu-Ultra scaffolds, adaptive orchestration, multi-agent learned 2606.21228: arXiv - Report Technical Fugu Sakana

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 045 - definitions technical Core

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Compute Vascular = توزيع الحساب لا حسب القوة العامة للنموذج بل حسب الوعاء: retrieve, verify, code, plan, synthesize.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: synthesis. verification, delegation, selection, model for interface API single sakana.ai/fugu-release: - Release Fugu Sakana

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

definitions technical Core - وحدة 046

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Orchestration Fugu-style = نظام يقرر query per كيف يستدعي agents of team ثم يصهرها في واجهة واحدة.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: catalog، model warnings, provider/privacy router, free routes, free openrouter.ai/collections/free-models: - Collection Models Free OpenRouter

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

definitions technical Core - وحدة 047

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Court Fusion = محكمة تقارن outputs وتفصل بين action، final and risk, mechanism, evidence, claim.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: NVIDIA NIM LLM API - docs.nvidia.com/nim/large-language-models/latest/api-reference.html: OpenAI-compatible inference NIM deployable for services.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback، rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

definitions technical Core - وحدة 048

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Memory Self-Reconstruction = ذاكرة تعيد بناء routing policy بناء على سبب الفشل أو النجاح لا على النص النهائي.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - ReAct: 2210.03629، with acting and reasoning interleaving tools/environments.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

definitions technical Core - وحدة 049

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperskill = مهارة فائقة قابلة للاستدعاء لها trigger, state, role, tools, verifier memory ,update and rule. retirement

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv: 2305.10601 - Tree of Thoughts: deliberate problem solving with large language models, multiple reasoning over search paths, and self-evaluation. backtracking

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

definitions technical Core - وحدة 050

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperloop = حلقة تشغيلية تعود إلى نفسها بتعديل في الذاكرة أو السياسة أو الأدلة، ليست تكرارا أعمى.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Self-Consistency :2203.11171 sampling multiple reasoning paths then selecting consistent answers.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

definitions technical Core - وحدة 051

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: MCD = مقياس فرق مركب يحدد متى يحتاج الطلب إلى routing بسيط أو court fusion كامل.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Reflexion: 2303.11366 verbal feedback and episodic memory for agents.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

definitions technical Core - وحدة 052

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Compute Vascular = توزيع الحساب لا حسب القوة العامة للنموذج بل حسب الوعاء: retrieve, verify, code, plan, synthesize.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - RAG: 2005.11401 parametric plus parametric non-memory for generation. knowledge-intensive

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

definitions technical Core - وحدة 053

expansion: integrated 2 Version حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Orchestration Fugu-style = نظام يقرر query per كيف يستدعي agents of team ثم يصهرها في واجهة واحدة.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: GraphRAG - Microsoft Research: arXiv + surveys: graph-structured retrieval, multi-hop communities, relations.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 054 - definitions technical Core

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Court Fusion = محكمة تقارن outputs وتفصل بين action، final and risk, mechanism, evidence, claim

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv: 2310.11511 - Self-RAG: learning to retrieve, generate, and critique through reflection، adaptive retrieval, critique generation, tokens.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch. من دون traces لا يوجد تعلم ولا مقارنة.

definitions technical Core - وحدة 055

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Memory Self-Reconstruction = ذاكرة تعيد بناء routing policy بناء على سبب الفشل أو النجاح لا على النص النهائي.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: split. speed-quality Fugu/Fugu-Ultra scaffolds, adaptive orchestration, multi-agent learned 2606.21228: arXiv - Report Technical Fugu Sakana

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

definitions technical Core - وحدة 056

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperskill = مهارة فائقة قابلة للاستدعاء لها trigger, state, role, tools, verifier memory, and update rule. retirement

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: Release Fugu Sakana - :sakana.ai/fugu-release API single model, for selection, delegation, verification, synthesis.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

definitions technical Core - وحدة 057

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperloop = حلقة تشغيلية تعود إلى نفسها بتعديل في الذاكرة أو السياسة أو الأدلة، ليست تكرارا أعمى.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: Collection Models Free OpenRouter - https://openrouter.ai/collections/free-models, free routes, free router/privacy warnings, model catalog.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 058 - definitions technical Core

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: MCD = مقياس فرق مركب يحدد متى يحتاج الطلب إلى routing بسيط أو court fusion كامل.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: NIM deployable for interface inference OpenAI-compatible docs.nvidia.com/nim/large-language-models/latest/api-reference.html: - API LLM NIM NVIDIA services.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

definitions technical Core - وحدة 059

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Compute Vascular = توزيع الحساب لا حسب القوة العامة للنموذج بل حسب الوعاء: retrieve, verify, code, plan, synthesize.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - ReAct :2210.03629. with acting and reasoning interleaving tools/environments.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

definitions technical Core - وحدة 060

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Orchestration Fugu-style = نظام يقرر query per كيف يستدعي agents of team ثم يصهرها في واجهة واحدة.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Thoughts of Tree: 2305.10601. backtracking، and self-evaluation paths, reasoning multiple over search

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

definitions technical Core - وحدة 061

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Court Fusion = محكمة تقارن outputs وتفصل بين action، final and risk, mechanism, evidence, claim

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Self-Consistency :2203.11171 sampling multiple reasoning paths then selecting consistent answers.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

definitions technical Core - وحدة 062

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Memory Self-Reconstruction = ذاكرة تعيد بناء routing policy بناء على سبب الفشل أو النجاح لا على النص النهائي.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Reflexion: 2303.11366 verbal feedback and episodic memory for agents.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

definitions technical Core - وحدة 063

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperskill = مهارة فائقة قابلة للاستدعاء لها state, trigger, tools, memory verifier, and update, rule. retirement

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - RAG: 2005.11401 .knowledge-intensive for memory non-parametric plus parametric generation

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 064 - definitions technical Core

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperloop = حلقة تشغيلية تعود إلى نفسها بتعديل في الذاكرة أو السياسة أو الأدلة، ليست تكرارا أعمى.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: relations. multi-hop communities, retrieval, graph-structured surveys: arXiv + GraphRAG Microsoft - GraphRAG

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

definitions technical Core - وحدة 065

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: MCD = مقياس فرق مركب يحدد متى يحتاج الطلب إلى routing بسيط أو court fusion كامل.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Self-RAG: 2310.11511، reflection through critique generation, retrieval, adaptive tokens.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

definitions technical Core - وحدة 066

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Compute Vascular = توزيع الحساب لا حسب القوة العامة للنموذج بل حسب الوعاء: retrieve, verify, code, plan, synthesize.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: split. speed-quality Fugu/Fugu-Ultra scaffolds, adaptive orchestration, multi-agent learned 2606.21228: arXiv - Report Technical Fugu Sakana

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

definitions technical Core - وحدة 067

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Orchestration Fugu-style = نظام يقرر query per كيف يستدعي agents of team ثم يصهرها في واجهة واحدة.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: Release Fugu Sakana - :sakana.ai/fugu-release single API for interface model, selection, delegation, verification, synthesis.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

Core technical definitions - وحدة 068

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Court Fusion = محكمة تقارن outputs وتفصل بين action، final and risk, mechanism, evidence, claim.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: Collection Models Free OpenRouter - https://openrouter.ai/collections/free-models, free routes, privacy/provider warnings، model، catalog

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

definitions technical Core - وحدة 069

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Memory Self-Reconstruction = ذاكرة تعيد بناء routing policy بناء على سبب الفشل أو النجاح لا على النص النهائي.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: NIM deployable for interface inference OpenAI-compatible docs.nvidia.com/nim/large-language-models/latest/api-reference.html: - API LLM NIM NVIDIA services.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, retirement. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

definitions technical Core - وحدة 070

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperskill = مهارة فائقة قابلة للاستدعاء لها trigger, state, role, tools, verifier memory ,update and rule. retirement

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - ReAct: 2210.03629. with acting and reasoning interleaving tools/environments

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

definitions technical Core - وحدة 071

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperloop = حلقة تشغيلية تعود إلى نفسها بتعديل في الذاكرة أو السياسة أو الأدلة، ليست تكرارا أعمى.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Thoughts of Tree 2305.10601: backtracking, and self-evaluation paths, reasoning multiple over search

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

definitions technical Core - وحدة 072

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: MCD = مقياس فرق مركب يحدد متى يحتاج الطلب إلى routing بسيط أو court fusion كامل.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Self-Consistency :2203.11171 sampling multiple reasoning paths then selecting consistent answers.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback أو rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

definitions technical Core - وحدة 073

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Compute Vascular = توزيع الحساب لا حسب القوة العامة للنموذج بل حسب الوعاء: retrieve, verify, code, plan, synthesize.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Reflexion: 2303.11366. verbal feedback and episodic memory for agents.

حقن 2v الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

definitions technical Core - وحدة 074

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Orchestration Fugu-style = نظام يقرر query per كيف يستدعي agents of team ثم يصهرها في واجهة واحدة.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - RAG :2005.11401 plus parametric non-parametric for memory knowledge-intensive generation.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

definitions technical Core - وحدة 075

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Court Fusion = محكمة تقارن outputs وتفصل بين action، final and risk, mechanism, evidence, claim

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: relations، multi-hop communities, retrieval, graph-structured surveys: arXiv + GraphRAG Microsoft - GraphRAG

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement، condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

definitions technical Core - وحدة 076

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Memory Self-Reconstruction = ذاكرة تعيد بناء routing policy بناء على سبب الفشل أو النجاح لا على النص النهائي.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Self-RAG: 2310.11511. reflection through critique generation, retrieval, adaptive tokens.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

definitions technical Core - وحدة 077

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperskill = مهارة فائقة قابلة للاستدعاء لها trigger, state, role, tools, memory verifier, and update, rule. retirement

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: arXiv - Report Technical Fugu Sakana :2606.21228, multi-agent learned scaffolds, adaptive orchestration, Fugu/Fugu-Ultra speed-quality split.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

definitions technical Core - وحدة 078

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Hyperloop = حلقة تشغيلية تعود إلى نفسها بتعديل في الذاكرة أو السياسة أو الأدلة، ليست تكرارا أعمى.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: synthesis. verification, delegation, selection, model for interface API single sakana.ai/fugu-release: - Release Fugu Sakana

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

definitions technical Core - وحدة 079

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: MCD = مقياس فرق مركب يحدد متى يحتاج الطلب إلى routing بسيط أو court fusion كامل.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: catalog، model warnings, provider/privacy router, free routes, free openrouter.ai/collections/free-models: - Collection Models Free OpenRouter

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

definitions technical Core - وحدة 080

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

التعريف المركزي: Compute Vascular = توزيع الحساب لا حسب القوة العامة للنموذج بل حسب الوعاء: retrieve, verify, code, plan, synthesize.

الوزن النظري لهذا المفهوم أنه يحول Hermes من واجهة تستدعي APIs إلى جهاز حاكم يملك ذاكرة أسباب، ومحكمة هدم، ومقياس تعقيد، وسياسة ترقية للمهارات. من غير هذا التعريف يصبح تعدد النماذج مجرد round-robin أو ensemble سطحي.

العنصر	الحد التقني
المدخل	طلب، سياق، قيود، مصادر، مخاطر
المعالجة	تفكيك، gnituor، تنفيذ، نقد، fusion
المخرج	جواب، أثر، سجل أدلة، تحديث ذاكرة
الفشل	إذا غاب verifier أو رتبة المعرفة أو أثر العالم

```
route_score = 0.25*domain + 0.20*reasoning + 0.15*context + 0.15*tools + 0.10*history - 0.10*risk
```

مصدر/إسناد: API LLM NIM NVIDIA - NIM deployable for interface inference OpenAI-compatible docs.nvidia.com/nim/large-language-models/latest/api-reference.html. services.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

ontology hyperskill AGI - وحدة 081

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Planning Deep لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Planning Deep أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

ontology hyperskill AGI - وحدة 082

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Repair Code لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Repair Code أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

ontology hyperskill AGI - وحدة 083

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Retrieval GraphRAG لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Retrieval GraphRAG أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

ontology hyperskill AGI - وحدة 084

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Critique Demolition لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Critique Demolition أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

ontology hyperskill AGI - وحدة 085

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Ranking Evidence لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Ranking Evidence أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

ontology hyperskill AGI - وحدة 086

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Execution Tool لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Execution Tool أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

ontology hyperskill AGI - وحدة 087

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Verification World-Effect لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Verification World-Effect أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

ontology hyperskill AGI - وحدة 088

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Rebuild Memory لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Rebuild Memory أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

ontology hyperskill AGI - وحدة 089

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Fallback Model لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Fallback Model أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

ontology hyperskill AGI - وحدة 090

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Red-Team Security لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Red-Team Security أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

ontology hyperskill AGI - وحدة 091

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Planning Deep لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Planning Deep أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

ontology hyperskill AGI - وحدة 092

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Repair Code لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Repair Code أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

ontology hyperskill AGI - وحدة 093

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Retrieval GraphRAG لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Retrieval GraphRAG أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

ontology hyperskill AGI - وحدة 094

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Critique Demolition لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Critique Demolition أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

ontology hyperskill AGI - وحدة 095

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Ranking Evidence لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Ranking Evidence أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

ontology hyperskill AGI - وحدة 096

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Execution Tool لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل .Hermes يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Execution Tool أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement .condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

ontology hyperskill AGI - وحدة 097

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Verification World-Effect لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Verification World-Effect أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

ontology hyperskill AGI - وحدة 098

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Rebuild Memory لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Rebuild Memory أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

ontology hyperskill AGI - وحدة 099

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Fallback Model لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Fallback Model أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

ontology hyperskill AGI - وحدة 100

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Red-Team Security لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Red-Team Security أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

ontology hyperskill AGI - وحدة 101

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Planning Deep لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Planning Deep أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

ontology hyperskill AGI - وحدة 102

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Repair Code لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Repair Code أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

ontology hyperskill AGI - وحدة 103

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Retrieval GraphRAG لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Retrieval GraphRAG أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

ontology hyperskill AGI - وحدة 104

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Critique Demolition لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Critique Demolition أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

ontology hyperskill AGI - وحدة 105

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Ranking Evidence لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Ranking Evidence أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

ontology hyperskill AGI - وحدة 106

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Execution Tool لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Execution Tool أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement condition. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

ontology hyperskill AGI - وحدة 107

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Verification World-Effect لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Verification World-Effect أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

ontology hyperskill AGI - وحدة 108

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Rebuild Memory لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Rebuild Memory أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

ontology hyperskill AGI - وحدة 109

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Fallback Model لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Fallback Model أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

ontology hyperskill AGI - وحدة 110

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Red-Team Security لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Red-Team Security أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

ontology hyperskill AGI - وحدة 111

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Planning Deep لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Planning Deep أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

ontology hyperskill AGI - وحدة 112

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Repair Code لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Repair Code أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

ontology hyperskill AGI - وحدة 113

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Retrieval GraphRAG لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Retrieval GraphRAG أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

ontology hyperskill AGI - وحدة 114

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Critique Demolition لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Critique Demolition أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

ontology hyperskill AGI - وحدة 115

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Ranking Evidence لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Ranking Evidence أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

ontology hyperskill AGI - وحدة 116

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Execution Tool لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل .Hermes يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Execution Tool أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement .condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

ontology hyperskill AGI - وحدة 117

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Verification World-Effect لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Verification World-Effect أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

ontology hyperskill AGI - وحدة 118

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Rebuild Memory لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Rebuild Memory أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

ontology hyperskill AGI - وحدة 119

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Fallback Model لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Fallback Model أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

ontology hyperskill AGI - وحدة 120

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Red-Team Security لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Red-Team Security أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

ontology hyperskill AGI - وحدة 121

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Planning Deep لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Planning Deep أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

ontology hyperskill AGI - وحدة 122

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Repair Code لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Repair Code أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role <- slice task <- evidence <- risk <- output <- verifier <- memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

ontology hyperskill AGI - وحدة 123

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Retrieval GraphRAG لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Retrieval GraphRAG أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

ontology hyperskill AGI - وحدة 124

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Critique Demolition لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Critique Demolition أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

ontology hyperskill AGI - وحدة 125

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Ranking Evidence لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Ranking Evidence أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

ontology hyperskill AGI - وحدة 126

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Execution Tool لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل .Hermes يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Execution Tool أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

ontology hyperskill AGI - وحدة 127

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Verification World-Effect لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Verification World-Effect أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

ontology hyperskill AGI - وحدة 128

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Rebuild Memory لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Rebuild Memory أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

ontology hyperskill AGI - وحدة 129

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Fallback Model لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Fallback Model أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

ontology hyperskill AGI - وحدة 130

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Red-Team Security لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Red-Team Security أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

ontology hyperskill AGI - وحدة 131

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Planning Deep لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Planning Deep أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

ontology hyperskill AGI - وحدة 132

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Repair Code لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Repair Code أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

ontology hyperskill AGI - وحدة 133

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Retrieval GraphRAG لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Retrieval GraphRAG أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

ontology hyperskill AGI - وحدة 134

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Critique Demolition لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Critique Demolition أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

ontology hyperskill AGI - وحدة 135

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Ranking Evidence لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Ranking Evidence أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

ontology hyperskill AGI - وحدة 136

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Execution Tool لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل .Hermes يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Execution Tool أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role <- slice task <- evidence <- risk <- output <- verifier <- memory <- patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement .condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

ontology hyperskill AGI - وحدة 137

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Verification World-Effect لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Verification World-Effect أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

ontology hyperskill AGI - وحدة 138

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Rebuild Memory لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Rebuild Memory أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

ontology hyperskill AGI - وحدة 139

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Fallback Model لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Fallback Model أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

ontology hyperskill AGI - وحدة 140

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Red-Team Security لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Red-Team Security أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

ontology hyperskill AGI - وحدة 141

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Planning Deep لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Planning Deep أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

ontology hyperskill AGI - وحدة 142

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Repair Code لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Repair Code أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

ontology hyperskill AGI - وحدة 143

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Retrieval GraphRAG لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Retrieval GraphRAG أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

ontology hyperskill AGI - وحدة 144

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Critique Demolition لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Critique Demolition أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

ontology hyperskill AGI - وحدة 145

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Ranking Evidence لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Ranking Evidence أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

ontology hyperskill AGI - وحدة 146

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Execution Tool لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Execution Tool أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, role, model, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

ontology hyperskill AGI - وحدة 147

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Verification World-Effect لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Verification World-Effect أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

ontology hyperskill AGI - وحدة 148

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Rebuild Memory لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Rebuild Memory أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

ontology hyperskill AGI - وحدة 149

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Fallback Model لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Fallback Model أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

ontology hyperskill AGI - وحدة 150

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المهارة الفائقة Red-Team Security لا تعرف بوصفها قدرة لغوية، بل بوصفها object تشغيلي داخل Hermes. يجب أن تحتوي على trigger واضح، حالة داخلية، حدود أمان، نموذج افتراضي، نموذج بديل، reifirev، وسجل تعلم.

حقل hyperskill	قيمة هندسية
trigger	عندما يظهر نمط Red-Team Security أو خطره
vessel primary	synthesizer / retriever / critic / coder / planner
court	operational + causal + factual
update memory	سبب الفوز، سبب الخطأ، patch rule
retirement	بعد تكرار فشل أو تجاوز تكلفة

الخرق هنا أن المهارة لا تبقى prompt محفوظا؛ تتحول إلى كيان معرفي يملك دورة حياة. كل hyperskill لها مولد، مستدعي، مقيم، ومقبرة. هذا يحمي Hermes من تراكم مهارات ميتة أو مزيفة.

```
Hyperskill = {trigger, state_schema, role_policy, tools, verifier, update_rule, retirement_condition}
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

patterns and laws Hyperloop - وحدة 151

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Delta-MCD هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

patterns and laws Hyperloop - وحدة 152

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Plan-Act-Observe هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

patterns and laws Hyperloop - وحدة 153

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Critique-Repair هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

patterns and laws Hyperloop - وحدة 154

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Retrieve-Verify هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

patterns and laws Hyperloop - وحدة 155

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Explore-Compress هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 156 - patterns and laws Hyperloop

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Hypothesis-Test هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

patterns and laws Hyperloop - وحدة 157

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Fallback-Sovereignty هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

Hyperloop patterns and laws - وحدة 158

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Skill-Upgrade Loop هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

Hyperloop patterns and laws - وحدة 159

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Delta-MCD هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 160 - patterns and laws Hyperloop

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Plan-Act-Observe هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

patterns and laws Hyperloop - وحدة 161

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Critique-Repair هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

patterns and laws Hyperloop - وحدة 162

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Retrieve-Verify هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

patterns and laws Hyperloop - وحدة 163

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Explore-Compress هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

patterns and laws Hyperloop - وحدة 164

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Hypothesis-Test هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

patterns and laws Hyperloop - وحدة 165

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Fallback-Sovereignty هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

Hyperloop patterns and laws - وحدة 166

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Skill-Upgrade هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

patterns and laws Hyperloop - وحدة 167

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Delta-MCD هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

Hyperloop patterns and laws - وحدة 168

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Plan-Act-Observe هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

patterns and laws Hyperloop - وحدة 169

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Critique-Repair هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

patterns and laws Hyperloop - وحدة 170

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Retrieve-Verify هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

Hyperloop patterns and laws - وحدة 171

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Explore-Compress Loop هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

patterns and laws Hyperloop - وحدة 172

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Hypothesis-Test هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

patterns and laws Hyperloop - وحدة 173

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Fallback-Sovereignty هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

patterns and laws Hyperloop - وحدة 174

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Skill-Upgrade هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

patterns and laws Hyperloop - وحدة 175

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Delta-MCD هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

patterns and laws Hyperloop - وحدة 176

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Plan-Act-Observe هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch، إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

patterns and laws Hyperloop - وحدة 177

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Critique-Repair هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

Hyperloop patterns and laws - وحدة 178

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Retrieve-Verify هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

وحدة 179 - patterns and laws Hyperloop

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Explore-Compress هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch، إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

patterns and laws Hyperloop - وحدة 180

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Hypothesis-Test هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

Hyperloop patterns and laws - وحدة 181

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Fallback-Sovereignty هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

patterns and laws Hyperloop - وحدة 182

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Skill-Upgrade هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

patterns and laws Hyperloop - وحدة 183

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Delta-MCD هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

patterns and laws Hyperloop - وحدة 184

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Plan-Act-Observe هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

patterns and laws Hyperloop - وحدة 185

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Critique-Repair هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

patterns and laws Hyperloop - وحدة 186

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Retrieve-Verify هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

patterns and laws Hyperloop - وحدة 187

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Explore-Compress هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

patterns and laws Hyperloop - وحدة 188

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Hypothesis-Test هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

patterns and laws Hyperloop - وحدة 189

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Fallback-Sovereignty هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

Hyperloop patterns and laws - وحدة 190

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Skill-Upgrade هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

patterns and laws Hyperloop - وحدة 191

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Delta-MCD هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

patterns and laws Hyperloop - وحدة 192

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Plan-Act-Observe هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

patterns and laws Hyperloop - وحدة 193

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Critique-Repair هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

patterns and laws Hyperloop - وحدة 194

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Retrieve-Verify هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

patterns and laws Hyperloop - وحدة 195

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Explore-Compress هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

Hyperloop patterns and laws - وحدة 196

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Test-Hypothesis Loop هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

patterns and laws Hyperloop - وحدة 197

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Fallback-Sovereignty هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

patterns and laws Hyperloop - وحدة 198

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Skill-Upgrade هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch، إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

patterns and laws Hyperloop - وحدة 199

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Delta-MCD هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

patterns and laws Hyperloop - وحدة 200

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Plan-Act-Observe هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

patterns and laws Hyperloop - وحدة 201

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Critique-Repair هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

patterns and laws Hyperloop - وحدة 202

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Retrieve-Verify هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 203 - patterns and laws Hyperloop

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Explore-Compress هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch، إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 204 - patterns and laws Hyperloop

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Hypothesis-Test هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

patterns and laws Hyperloop - وحدة 205

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Fallback-Sovereignty هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

Hyperloop patterns and laws - وحدة 206

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Skill-Upgrade هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

patterns and laws Hyperloop - وحدة 207

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Delta-MCD هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 208 - patterns and laws Hyperloop

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Plan-Act-Observe هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

Hyperloop patterns and laws - وحدة 209

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Critique-Repair هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 210 - patterns and laws Hyperloop

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Retrieve-Verify هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

patterns and laws Hyperloop - وحدة 211

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Explore-Compress هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

patterns and laws Hyperloop - وحدة 212

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Hypothesis-Test هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

patterns and laws Hyperloop - وحدة 213

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Fallback-Sovereignty هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

patterns and laws Hyperloop - وحدة 214

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Skill-Upgrade هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

patterns and laws Hyperloop - وحدة 215

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Delta-MCD هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

patterns and laws Hyperloop - وحدة 216

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Plan-Act-Observe هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

patterns and laws Hyperloop - وحدة 217

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Critique-Repair هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 218 - patterns and laws Hyperloop

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Retrieve-Verify هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

patterns and laws Hyperloop - وحدة 219

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Explore-Compress هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 220 - patterns and laws Hyperloop

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

Loop Hypothesis-Test هو نمط حلقي لا يكرر نفسه إلا بشرط: أن يعود بمعلومة تغير routing أو الذاكرة أو المقياس أو النموذج. الحلقة التي لا تغير شيئا ليست hyperloop بل استنزاف tokens.

- مرحلة الفرق: استخراج ما تغير أو ما لا نعرفه.
- مرحلة الوعاء: اختيار نموذج أو أداة حسب نوع الفرق.
- مرحلة المحكمة: البحث عن التناقض أو الفجوة السببية.
- مرحلة الأثر: اختبار صغير أو معيار داخلي أو تحويل إلى قاعدة.
- مرحلة الذاكرة: حفظ rule لا حفظ كلام طويل.

خطر الحلقة	كابح v2
reflection infinite	حد أقصى للدورات + مخرج إجباري
illusion consensus	مسار مضاد مستقل
drift tool	validation schema + sandbox
over-routing	Dial Latency-Quality

```
while uncertainty > threshold and budget.ok():
    path = router.select_vessel(delta, mcd)
    result = path.run()
    verdict = court.attack(result)
    memory.patch(verdict.rule)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

router fusion Hermes-Fugu - وحدة 221

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: FAST. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وsemreH v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Release Fugu Sakana - sakana.ai/fugu-release/: verification, delegation, selection, model for interface API single synthesis.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

router fusion Hermes-Fugu - وحدة 222

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: DEEP. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وsemreH v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Collection Models Free OpenRouter - https://openrouter.ai/collections/free-models, free routes, provider/privacy router, model warnings, catalog.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

router fusion Hermes-Fugu - وحدة 223

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: CODE_FUSION. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علامة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: NVIDIA NIM LLM API - docs.nvidia.com/nim/large-language-models/latest/api-reference.html: OpenAI-compatible inference interface deployable NIM services.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

router fusion Hermes-Fugu - وحدة 224

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: MULTIMODAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - ReAct: 2210.03629 with acting and reasoning interleaving tools/environments.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

router fusion Hermes-Fugu - وحدة 225

Version 2 :integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: .RESEARCH في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علامة .task-risk-evidence-budget

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Thoughts of Tree :2305.10601 over search paths, reasoning multiple and self-evaluation backtracking.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict .patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement .condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict .patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

router fusion Hermes-Fugu - وحدة 226

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SECURITY. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علامة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Self-Consistency :2203.11171 sampling multiple reasoning paths then consistent selecting answers.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية من: task, model, role, evidence, verdict, patch. traces من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

router fusion Hermes-Fugu - وحدة 227

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SOVEREIGN_LOCAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Reflexion: 2303.11366 verbal feedback and episodic memory for agents.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 228 - router fusion Hermes-Fugu

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: FAST. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وsemreH v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - RAG: 2005.11401، knowledge-intensive for memory non-parametric plus parametric generation.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement، condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement، condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

router fusion Hermes-Fugu - وحدة 229

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: DEEP. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation?
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis?

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: GraphRAG - Microsoft + arXiv: surveys, graph-structured retrieval, multi-hop communities, relations.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

router fusion Hermes-Fugu - وحدة 230

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: CODE_FUSION. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Self-RAG: 2310.11511. reflection through critique generation, retrieval, adaptive tokens.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

router fusion Hermes-Fugu - وحدة 231

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: MULTIMODAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: split. speed-quality Fugu/Fugu-Ultra scaffolds, adaptive orchestration, multi-agent learned 2606.21228: arXiv - Report Technical Fugu Sakana

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, role, model, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

router fusion Hermes-Fugu - وحدة 232

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: RESEARCH. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علامة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Release Fugu Sakana - sakana.ai/fugu-release/: synthesis, verification, delegation, selection, model for interface API single.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

router fusion Hermes-Fugu - وحدة 233

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SECURITY. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: catalog, model warnings, provider/privacy router, free routes, free openrouter.ai/collections/free-models: - Collection Models Free OpenRouter

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

router fusion Hermes-Fugu - وحدة 234

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SOVEREIGN_LOCAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: NVIDIA NIM API LLM - docs.nvidia.com/nim/large-language-models/latest/api-reference.html: OpenAI-compatible interface inference deployable NIM services.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

router fusion Hermes-Fugu - وحدة 235

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: FAST. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - ReAct: 2210.03629 with acting and reasoning interleaving tools/environments.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

router fusion Hermes-Fugu - وحدة 236

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: DEEP. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وsemreH v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation?
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis?

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Thoughts of Tree 2305.10601: backtracking, and self-evaluation paths, reasoning multiple over search

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية من: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

router fusion Hermes-Fugu - وحدة 237

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: CODE_FUSION. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Self-Consistency: 2203.11171، sampling multiple reasoning paths then consistent selecting answers.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

router fusion Hermes-Fugu - وحدة 238

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: MULTIMODAL. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Reflexion: 2303.11366 episodic and feedback verbal for memory agents.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

router fusion Hermes-Fugu - وحدة 239

Version integrated 2 :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: RESEARCH. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علامة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - RAG :2005.11401 plus parametric non-parametric memory for knowledge-intensive generation.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

router fusion Hermes-Fugu - وحدة 240

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SECURITY. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: GraphRAG - Microsoft + arXiv: surveys, graph-structured retrieval, multi-hop communities, relations.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

router fusion Hermes-Fugu - وحدة 241

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SOVEREIGN_LOCAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Self-RAG - arXiv: 2310.11511، reflection through critique generation, retrieval, adaptive tokens.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

router fusion Hermes-Fugu - وحدة 242

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: FAST. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وsemreH v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: split. speed-quality Fugu/Fugu-Ultra scaffolds, adaptive orchestration, multi-agent learned 2606.21228: arXiv - Report Technical Fugu Sakana

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

router fusion Hermes-Fugu - وحدة 243

Version 2 :integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: DEEP. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وsemreH v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Release Fugu Sakana - sakana.ai/fugu-release/: API single interface for model, selection, delegation, verification, synthesis.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

router fusion Hermes-Fugu - وحدة 244

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: CODE_FUSION. في Fugu-style لا تثبت workflow واحدا؛ تنتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Collection Models Free OpenRouter - https://openrouter.ai/collections/free-models, free routes, provider/privacy router, model warnings, catalog.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

router fusion Hermes-Fugu - وحدة 245

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: MULTIMODAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: NVIDIA NIM API LLM - docs.nvidia.com/nim/large-language-models/latest/api-reference.html: OpenAI-compatible interface inference deployable NIM services.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل planner داخل وعاء Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

router fusion Hermes-Fugu - وحدة 246

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: RESEARCH. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - ReAct: 2210.03629 with acting and reasoning interleaving tools/environments.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية من: task, model, role, evidence, verdict, patch. لا يوجد تعلم ولا مقارنة من دون traces.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

router fusion Hermes-Fugu - وحدة 247

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SECURITY. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علامة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Thoughts of Tree 2305.10601: search over multiple reasoning, backtracking, and self-evaluation paths

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

router fusion Hermes-Fugu - وحدة 248

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SOVEREIGN_LOCAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علامة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Self-Consistency: 2203.11171 multiple sampling reasoning paths then selecting consistent answers.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

router fusion Hermes-Fugu - وحدة 249

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: FAST. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Reflexion: 2303.11366 episodic and feedback verbal for memory agents.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

router fusion Hermes-Fugu - وحدة 250

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: DEEP. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وsemreH v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - RAG: 2005.11401 non-parametric plus parametric memory for knowledge-intensive generation.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

router fusion Hermes-Fugu - وحدة 251

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: CODE_FUSION. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: GraphRAG - Microsoft + arXiv: graph-structured surveys, retrieval, multi-hop communities, relations.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

router fusion Hermes-Fugu - وحدة 252

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: MULTIMODAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Self-RAG: 2310.11511. reflection through critique generation, retrieval, adaptive tokens.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

router fusion Hermes-Fugu - وحدة 253

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: RESEARCH. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: split. speed-quality Fugu/Fugu-Ultra scaffolds, adaptive orchestration, multi-agent learned 2606.21228: arXiv - Report Technical Fugu Sakana

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

router fusion Hermes-Fugu - وحدة 254

Version 2 :integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SECURITY. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Release Fugu Sakana - sakana.ai/fugu-release/, API single interface for model, selection, delegation, verification, synthesis.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

router fusion Hermes-Fugu - وحدة 255

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SOVEREIGN_LOCAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Collection Models Free OpenRouter - https://openrouter.ai/collections/free-models, free routes, free router/privacy provider, warnings model, catalog.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

router fusion Hermes-Fugu - وحدة 256

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: FAST. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: NVIDIA NIM LLM API - docs.nvidia.com/nim/large-language-models/latest/api-reference.html: OpenAI-compatible inference interface for NIM deployable services.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

router fusion Hermes-Fugu - وحدة 257

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: DEEP. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وsemreH v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - ReAct: 2210.03629. with acting and reasoning interleaving tools/environments

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

router fusion Hermes-Fugu - وحدة 258

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: CODE_FUSION. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Thoughts of Tree 2305.10601: backtracking, and self-evaluation paths, reasoning multiple over search

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

router fusion Hermes-Fugu - وحدة 259

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: MULTIMODAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علامة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Self-Consistency: 2203.11171 multiple sampling then selecting consistent answers.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

router fusion Hermes-Fugu - وحدة 260

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: .RESEARCH في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة .task-risk-evidence-budget

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: agents. for memory episodic and feedback verbal 2303.11366: arXiv - Reflexion

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

router fusion Hermes-Fugu - وحدة 261

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SECURITY. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - RAG: 2005.11401. knowledge-intensive for memory non-parametric plus parametric generation.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

router fusion Hermes-Fugu - وحدة 262

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SOVEREIGN_LOCAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: relations. multi-hop communities, retrieval, graph-structured surveys: arXiv + GraphRAG Microsoft - GraphRAG

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

router fusion Hermes-Fugu - وحدة 263

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: FAST. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Self-RAG: 2310.11511. reflection through critique generation, retrieval, adaptive tokens.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

router fusion Hermes-Fugu - وحدة 264

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: DEEP. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وsemreH v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: split. speed-quality Fugu/Fugu-Ultra scaffolds, adaptive orchestration, multi-agent learned 2606.21228: arXiv - Report Technical Fugu Sakana

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

router fusion Hermes-Fugu - وحدة 265

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: CODE_FUSION. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Release Fugu Sakana - https://sakana.ai/fugu-release/: interface API single sakana, model, selection, delegation, verification, synthesis.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 266 - router fusion Hermes-Fugu

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: MULTIMODAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Collection Models Free OpenRouter - https://openrouter.ai/collections/free-models, free routes, provider/privacy router, model warnings, catalog.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

router fusion Hermes-Fugu - وحدة 267

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: RESEARCH. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: NVIDIA NIM API LLM - docs.nvidia.com/nim/large-language-models/latest/api-reference.html: OpenAI-compatible inference interface deployable NIM services.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

router fusion Hermes-Fugu - وحدة 268

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SECURITY. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - ReAct: 2210.03629 with acting and reasoning interleaving tools/environments.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

router fusion Hermes-Fugu - وحدة 269

Version 2 :integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SOVEREIGN_LOCAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Thoughts of Tree: 2305.10601. reasoning multiple over search backtracking, and self-evaluation paths

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

router fusion Hermes-Fugu - وحدة 270

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: FAST. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Self-Consistency :2203.11171 sampling multiple reasoning paths then selecting consistent answers.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

router fusion Hermes-Fugu - وحدة 271

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: DEEP. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Reflexion: 2303.11366 verbal feedback and episodic memory for agents.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

router fusion Hermes-Fugu - وحدة 272

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: CODE_FUSION. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علامة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - RAG: 2005.11401. knowledge-intensive for memory non-parametric plus parametric generation.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

router fusion Hermes-Fugu - وحدة 273

Version 2 :integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: MULTIMODAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: relations. multi-hop communities, retrieval, graph-structured surveys: arXiv + GraphRAG Microsoft - GraphRAG

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

router fusion Hermes-Fugu - وحدة 274

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: RESEARCH. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Self-RAG - arXiv :2310.11511 reflection through critique generation, retrieval, adaptive tokens.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

router fusion Hermes-Fugu - وحدة 275

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SECURITY. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Hermes Fugu Technical Report - arXiv: 2606.21228، speed-quality Fugu/Fugu-Ultra scaffolds, adaptive orchestration, multi-agent learned split.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

router fusion Hermes-Fugu - وحدة 276

Version 2 :integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: .SOVEREIGN_LOCAL في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علامة .task-risk-evidence-budget

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Release Fugu Sakana - sakana.ai/fugu-release/: API single interface for model, selection, delegation, verification, synthesis.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

router fusion Hermes-Fugu - وحدة 277

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: FAST. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Collection Models Free OpenRouter - https://openrouter.ai/collections/free-models: catalog, model warnings, provider/privacy router, free routes, free

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

router fusion Hermes-Fugu - وحدة 278

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: DEEP في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: NVIDIA NIM LLM API - docs.nvidia.com/nim/large-language-models/latest/api-reference.html: OpenAI-compatible inference interface deployable NIM services.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

router fusion Hermes-Fugu - وحدة 279

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: CODE_FUSION. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - ReAct: 2210.03629 with acting and reasoning interleaving tools/environments.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

router fusion Hermes-Fugu - وحدة 280

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: MULTIMODAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Thoughts of Tree 2305.10601: backtracking, and self-evaluation paths, reasoning multiple over search.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

router fusion Hermes-Fugu - وحدة 281

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: RESEARCH. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Self-Consistency :2203.11171 sampling multiple reasoning paths then consistent selecting answers.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

router fusion Hermes-Fugu - وحدة 282

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SECURITY. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Reflexion: 2303.11366 verbal feedback and episodic memory for agents.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

router fusion Hermes-Fugu - وحدة 283

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SOVEREIGN_LOCAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - RAG: 2005.11401. knowledge-intensive for memory non-parametric plus parametric generation.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

router fusion Hermes-Fugu - وحدة 284

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: FAST. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وsemreH v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: GraphRAG - Microsoft + arXiv: surveys, graph-structured retrieval, multi-hop communities, relations.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

router fusion Hermes-Fugu - وحدة 285

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: DEEP. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - Self-RAG: 2310.11511. reflection through critique generation, retrieval, adaptive tokens.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

router fusion Hermes-Fugu - وحدة 286

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: CODE_FUSION. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علامة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: split. speed-quality Fugu/Fugu-Ultra scaffolds, adaptive orchestration, multi-agent learned 2606.21228: arXiv - Report Technical Fugu Sakana

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

router fusion Hermes-Fugu - وحدة 287

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: MULTIMODAL. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Release Fugu Sakana - sakana.ai/fugu-release/: verification, delegation, selection, model for interface API single synthesis.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

router fusion Hermes-Fugu - وحدة 288

Version 2 :integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: RESEARCH. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علامة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: Collection Models Free OpenRouter - https://openrouter.ai/collections/free-models, free routes, privacy router, provider warnings, model catalog.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

router fusion Hermes-Fugu - وحدة 289

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SECURITY. في Fugu-style لا نثبت workflow واحدا، ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: NVIDIA NIM API LLM: - docs.nvidia.com/nim/large-language-models/latest/api-reference.html OpenAI-compatible interface inference deployable NIM services.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

router fusion Hermes-Fugu - وحدة 290

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

وضع Hermes-Fugu في هذه الوحدة: SOVEREIGN_LOCAL. في Fugu-style لا نثبت workflow واحدا؛ ننتج scaffold لكل طلب. الفرق بين router العادي وHermes v2 أن الاختيار لا يعتمد على اسم النموذج، بل على علاقة task-risk-evidence-budget.

layer Decision	Question
complexity	هل المسألة تحتاج أكثر من مسار؟
risk	هل الخطأ مكلف أو أمني أو معرفي؟
evidence	هل يلزم RAG/GraphRAG/citation؟
tools	هل يلزم تنفيذ أو قراءة ملفات أو كود؟
fusion	هل توجد مخرجات متعارضة تحتاج synthesis؟

المبدأ الحاكم: نموذج واحد للسرعة، نموذج + verifier للمتوسط، فريق متعدد للنادر الصعب. كل تشغيل زائد بلا مبرر يضعف النظام لأنه يرفع latency ويزيد نقاط الفشل.

```
if mcd < 0.35: use FAST
elif risk.high or domain.hard: use DEEP + COURT
elif task.code: use CODE_FUSION
else: use PLANNER + WORKER + VERIFIER
```

مصدر/إسناد: arXiv - ReAct: 2210.03629 with acting and reasoning interleaving tools/environments.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 291 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن route free OpenRouter vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: backtracking. and self-evaluation paths, reasoning multiple over search 2305.10601: arXiv - Thoughts of Tree

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

وحدة 292 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor OpenRouter direct model slug ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: answers. consistent selecting then paths reasoning multiple sampling 2203.11171: arXiv - Self-Consistency

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 293 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor NVIDIA NIM hosted endpoint ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: agents. for memory episodic and feedback verbal 2303.11366: arXiv - Reflexion

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 294 - layer provider OpenRouter/NVIDIA

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن self-host NIM NVIDIA vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا Hermes private. v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - RAG :11401.2005 plus parametric non-parametric memory for knowledge-intensive generation.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 295 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor fallback vLLM/TGI Local ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: relations. multi-hop communities, retrieval, graph-structured surveys: arXiv + GraphRAG Microsoft - GraphRAG

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 296 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن policy privacy Hybrid vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا Hermes private. v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - Self-RAG: 2310.11511 tokens. reflection through critique generation, retrieval, adaptive

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 297 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor OpenRouter free route ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .v2 Hermes private يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - Report Technical Fugu Sakana :2606.21228 multi-agent learned, adaptive orchestration, scaffolds Fugu/Fugu-Ultra speed-quality. split.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 298 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن OpenRouter vendor. slug model direct ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: Release Fugu Sakana - https://sakana.ai/fugu-release/: synthesis. verification, delegation, selection, model for interface API single

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

وحدة 299 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن endpoint hosted NIM NVIDIA vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: catalog. model warnings, provider/privacy router, free routes, free openrouter.ai/collections/free-models: - Collection Models Free OpenRouter

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 300 - layer provider OpenRouter/NVIDIA

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن self-host NIM NVIDIA vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا v2 Hermes private. يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: NIM deployable for interface inference OpenAI-compatible docs.nvidia.com/nim/large-language-models/latest/api-reference.html: - API LLM NIM NVIDIA services.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 301 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor fallback vLLM/TGI Local ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - ReAct: 2210.03629 with acting and reasoning interleaving tools/environments.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: ,task ,model ,role ,evidence ,verdict .patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. ,retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

وحدة 302 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن policy privacy Hybrid vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا Hermes private. v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: backtracking. and self-evaluation paths, reasoning multiple over search 2305.10601: arXiv - Thoughts of Tree

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 303 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن route free OpenRouter vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, license, cost, latency, capacity, and reproducibility.

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى free hosted routes.

مصدر/إسناد: arXiv - Self-Consistency: 2203.11171 sampling multiple reasoning paths consistent selecting then answers.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 304 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor OpenRouter direct model slug ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: agents. for memory episodic and feedback verbal 2303.11366: arXiv - Reflexion

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 305 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن vendor. NIM NVIDIA hosted endpoint ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - RAG :2005.11401 plus parametric non-parametric memory for generation. knowledge-intensive

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 306 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن self-host NIM NVIDIA vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا Hermes private. v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: relations. multi-hop communities, retrieval, graph-structured surveys: arXiv + GraphRAG Microsoft - GraphRAG

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 307 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor fallback vLLM/TGI Local ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - Self-RAG: 2310.11511 tokens. reflection through critique generation, retrieval, adaptive

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 308 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن policy privacy Hybrid vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: Report Technical Fugu Sakana - arXiv :2606.21228 scaffolds, adaptive orchestration, multi-agent learned Fugu/Fugu-Ultra speed-quality split.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

وحدة 309 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن route free OpenRouter vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: Release Fugu Sakana - :synthesis. verification, delegation, selection, model for interface API single sakana.ai/fugu-release

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 310 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor OpenRouter direct model slug ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: catalog. model warnings, provider/privacy router, free routes, free openrouter.ai/collections/free-models: - Collection Models Free OpenRouter

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, ,role ,model ,evidence ,verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك .retirement condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 311 - layer provider OpenRouter/NVIDIA

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن endpoint hosted NIM NVIDIA vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا Hermes private. v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: NIM deployable for interface inference OpenAI-compatible docs.nvidia.com/nim/large-language-models/latest/api-reference.html: - API LLM NIM NVIDIA services.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

وحدة 312 - layer provider OpenRouter/NVIDIA

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن self-host NIM NVIDIA vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا Hermes private. v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - ReAct: 2210.03629 tools/environments. with acting and reasoning interleaving

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 313 - layer provider OpenRouter/NVIDIA

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor Local vLLM/TGI fallback ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: backtracking. and self-evaluation paths, reasoning multiple over search 2305.10601: arXiv - Thoughts of Tree

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 314 - layer provider OpenRouter/NVIDIA

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن policy privacy Hybrid vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - Self-Consistency: 2203.11171 sampling multiple reasoning paths then consistent selecting answers.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 315 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor OpenRouter free route ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: agents. for memory episodic and feedback verbal 2303.11366: arXiv - Reflexion

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 316 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن OpenRouter vendor. slug model direct ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - RAG :2005.11401 plus parametric non-parametric memory for generation. knowledge-intensive

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 317 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن vendor. endpoint hosted NIM NVIDIA ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: GraphRAG - Microsoft + arXiv: surveys graph-structured, retrieval, communities multi-hop, relations.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 318 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن self-host NIM NVIDIA vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .Hermes private v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - Self-RAG: 2310.11511 tokens. reflection through critique generation, retrieval, adaptive

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

وحدة 319 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن vendor. Local vLLM/TGI fallback ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: Report Technical Fugu Sakana - arXiv: 2606.21228، scaffolds Fugu/Fugu-Ultra speed-quality، adaptive orchestration، multi-agent learned، split.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 320 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor Hybrid privacy policy ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, license, cost, latency, capacity, and reproducibility.

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: Release Fugu Sakana - sakana.ai/fugu-release/: synthesis. verification, delegation, selection, model for interface API single

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 321 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor OpenRouter free route ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .v2 Hermes private يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: catalog. model warnings, provider/privacy router, free routes, free openrouter.ai/collections/free-models: - Collection Models Free OpenRouter

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

وحدة 322 - layer provider OpenRouter/NVIDIA

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن OpenRouter vendor direct model slug ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا Hermes private. v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: NIM deployable for interface inference OpenAI-compatible docs.nvidia.com/nim/large-language-models/latest/api-reference.html: - API LLM NIM NVIDIA services.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rule. rollback هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 323 - layer provider OpenRouter/NVIDIA

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن vendor. endpoint hosted NIM NVIDIA ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. v2 Hermes يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - ReAct: 2210.03629 tools/environments. with acting and reasoning interleaving

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 324 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن self-host NIM NVIDIA vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .Hermes private v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: backtracking. and self-evaluation paths, reasoning multiple over search 2305.10601: arXiv - Thoughts of Tree

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 325 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor fallback vLLM/TGI Local ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: answers. consistent selecting then paths reasoning multiple sampling 2203.11171: arXiv - Self-Consistency

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 326 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن policy privacy Hybrid vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - Reflexion: 2303.11366. episodic and feedback verbal for memory agents.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 327 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor OpenRouter free route ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - RAG :2005.11401 generation. knowledge-intensive for memory non-parametric plus parametric

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 328 - layer provider OpenRouter/NVIDIA

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن OpenRouter vendor. slug model direct ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: GraphRAG - GraphRAG Microsoft + arXiv surveys: graph-structured retrieval, multi-hop communities, relations.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

وحدة 329 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن endpoint hosted NIM NVIDIA vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا Hermes private. v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - Self-RAG: 2310.11511 reflection through critique generation, retrieval, adaptive tokens.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 330 - layer provider OpenRouter/NVIDIA

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن self-host NIM NVIDIA vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا Hermes private. v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: Sakana Fugu Technical Report - arXiv: 2606.21228، multi-agent learned, adaptive orchestration, scaffolds Fugu/Fugu-Ultra speed-quality. split

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, role, model, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 331 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor fallback vLLM/TGI Local ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .v2 Hermes private يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: synthesis. verification, delegation, selection, model for interface API single sakana.ai/fugu-release: - Release Fugu Sakana

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

وحدة 332 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن policy privacy Hybrid vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا Hermes private. v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: catalog. model warnings, provider/privacy router, free routes, free openrouter.ai/collections/free-models: - Collection Models Free OpenRouter

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 333 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن vendor. OpenRouter free route ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى free hosted routes.

مصدر/إسناد: NVIDIA NIM LLM API - docs.nvidia.com/nim/large-language-models/latest/api-reference.html: OpenAI-compatible inference interface for deployable NIM services.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 334 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor OpenRouter direct model slug ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - ReAct: 2210.03629 with acting and reasoning interleaving tools/environments.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 335 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن vendor. endpoint hosted NIM NVIDIA ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - Thoughts of Tree 2305.10601: search over multiple reasoning, and self-evaluation paths. backtracking

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 336 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .self-host NIM NVIDIA vendor ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .v2 Hermes private يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: answers. consistent selecting then paths reasoning multiple sampling 2203.11171: arXiv - Self-Consistency

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 337 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن vendor. fallback vLLM/TGI Local ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا Hermes private. v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - Reflexion: 2303.11366. episodic and feedback verbal for memory agents.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 338 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن policy privacy Hybrid vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .Hermes private v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - RAG :2005.11401 plus parametric non-parametric memory for generation. knowledge-intensive

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

وحدة 339 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن route free OpenRouter vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .Hermes private v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: relations. multi-hop communities, retrieval, graph-structured surveys: arXiv + GraphRAG Microsoft - GraphRAG

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 340 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor OpenRouter direct model slug ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: tokens. reflection through critique generation, retrieval, adaptive 2310.11511: arXiv - Self-RAG

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, role, model, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك .condition retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 341 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن vendor. endpoint hosted NIM NVIDIA ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: Report Technical Fugu Sakana :arXiv 2606.21228. multi-agent learned, adaptive orchestration, scaffolds Fugu/Fugu-Ultra. speed-quality split.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

وحدة 342 - layer provider OpenRouter/NVIDIA

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن vendor. NIM NVIDIA self-host ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا Hermes private. v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: Release Fugu Sakana - :sakana.ai/fugu-release synthesis. verification, delegation, selection, model for interface API single

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 343 - layer provider OpenRouter/NVIDIA

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن vendor. Local vLLM/TGI fallback ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: Collection Models Free OpenRouter - https://openrouter.ai/collections/free-models: catalog. model warnings, provider/privacy router, free routes, free

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 344 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن policy privacy Hybrid vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, license, cost, latency, capacity, and reproducibility.

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى free hosted routes.

مصدر/إسناد: NVIDIA NIM LLM API - docs.nvidia.com/nim/large-language-models/latest/api-reference.html: OpenAI-compatible inference interface for deployable NIM services.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 345 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor OpenRouter free route ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .v2 Hermes private يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - ReAct :2210.03629 tools/environments. with acting and reasoning interleaving

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 346 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن OpenRouter vendor. slug model direct ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: backtracking. and self-evaluation paths, reasoning multiple over search 2305.10601: arXiv - Thoughts of Tree

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 347 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن vendor. endpoint hosted NIM NVIDIA ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا private. Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - Self-Consistency: 2203.11171 sampling multiple reasoning paths then consistent selecting answers.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 348 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن self-host NIM NVIDIA vendor. ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا Hermes private. v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - Reflexion :2303.11366 episodic and feedback verbal for memory agents.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

وحدة 349 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor Local vLLM/TGI fallback ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: reproducibility. and license, cost, latency, capacity, privacy,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: arXiv - RAG :2005.11401 generation. knowledge-intensive for memory non-parametric plus parametric

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 350 - layer provider OpenRouter/NVIDIA

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

طبقة provider في v2 تفصل intelligence عن .vendor Hybrid privacy policy ليس مجرد tniopdne؛ هو وعاء سيادي له حدود: privacy, reproducibility. and license, cost, latency, capacity,

field Provider	Policy
hosted-free	code client or secrets no only; prototype
model direct	capability specific requires role when use
router free	exploration low-risk for use
NIM/self-host	production stable and privacy for use
audit	failure role, timestamp, provider, model, record

```
ProviderCall = {base_url, api_key_ref, model, role, privacy_tier, timeout, retry, audit_id}
```

القاعدة: open/free لا يعني unlimited ولا .private Hermes v2 يجب أن يملك policy تمنع إرسال مفاتيح IPA، كود زبناء، أو وثائق حساسة إلى routes. free hosted

مصدر/إسناد: relations. multi-hop communities, retrieval, graph-structured surveys: arXiv + GraphRAG Microsoft - GraphRAG

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك .retirement condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

Moussaif engineering AGI and theories - وحدة 351

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الحاكمية الوعائية متعددة النماذج / Multi-Model Vascular Governance Theory. لا تكون القدرة في نموذج واحد، بل في طريقة توزيع الانتباه والحساب والذاكرة والمحكمة بين أوعية متخصصة.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

352 وحدة - engineering AGI and theories Moussaif

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية المهارة الفائقة كبرنامج تشغيل معرفي / .Hyperskill-as-Cognitive-Operating-Program المهارة الفائقة ليست tpmorp؛ إنها trigger + state + tools + verifier + rule update + retirement. condition

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار .routing
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

engineering AGI and theories Moussaif - وحدة 353

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الحلقة الفائقة المصعدة / Hyperloop Ascending Theory. كل حلقة تفكير يجب أن تعود بنتيجة تغير قاعدة المعرفة أو طريقة التوجيه، لا بمجرد إجابة عابرة.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

engineering AGI and theories Moussaif - وحدة 354

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية محكمة الهدم التوليدية / Generative Court Demolition Theory. النقد لا يقتل النموذج؛ يحول العيب إلى تعريف أو أداة أو محور بحث أو تخفيض رتبة معرفية.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

Moussaif engineering AGI and theories - وحدة 355

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الانتروبيا الاختراعية المضبوطة / Governed Inventive Entropy Theory. الابتكار الأعلى يأتي من رفع مساحة البدائل ثم كبحها بمراتب الدليل والاختبار والأثر العالمي.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

Moussaif engineering AGI and theories - وحدة 356

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الحاكمية الوعائية متعددة النماذج / Multi-Model Vascular Governance Theory. لا تكون القدرة في نموذج واحد، بل في طريقة توزيع الانتباه والحساب والذاكرة والمحكمة بين أوعية متخصصة.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

Moussaif engineering AGI and theories - وحدة 357

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية المهارة الفائقة كبرنامج تشغيل معرفي / Hyperskill-as-Cognitive-Operating-Program. المهارة الفائقة ليست tpmorp؛ إنها trigger + state + tools + verifier + rule update + retirement. condition

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

Moussaif engineering AGI and theories - وحدة 358

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الحلقة الفائقة المصعدة / Hyperloop Ascending .Theory كل حلقة تفكير يجب أن تعود بنتيجة تغير قاعدة المعرفة أو طريقة التوجيه، لا بمجرد إجابة عابرة.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار .routing
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition .retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition .retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

Moussaif engineering AGI and theories - وحدة 359

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية محكمة الهدم التوليدية / Generative Court Demolition Theory. النقد لا يقتل النموذج؛ يحول العيب إلى تعريف أو أداة أو محور بحث أو تخفيض رتبة معرفية.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

Moussaif engineering AGI and theories - وحدة 360

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الانتروبيا الاختراعية المضبوطة / Entropy Inventive Governed Theory. الابتكار الأعلى يأتي من رفع مساحة البدائل ثم كبحها بمراتب الدليل والاختبار والأثر العالمي.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

Moussaif engineering AGI and theories - وحدة 361

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الحاكمية الوعائية متعددة النماذج / Multi-Model Vascular Governance Theory. لا تكون القدرة في نموذج واحد، بل في طريقة توزيع الانتباه والحساب والذاكرة والمحكمة بين أوعية متخصصة.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة LLM systems عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

Moussaif engineering AGI and theories - وحدة 362

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية المهارة الفائقة كبرنامج تشغيل معرفي / .Hyperskill-as-Cognitive-Operating-Program المهارة الفائقة ليست tpmorp؛ إنها trigger + state + tools + verifier + rule update + retirement. condition

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار .routing
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

Moussaif engineering AGI and theories - وحدة 363

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الحلقة الفائقة المصعدة / Hyperloop Ascending Theory. كل حلقة تفكير يجب أن تعود بنتيجة تغير قاعدة المعرفة أو طريقة التوجيه، لا بمجرد إجابة عابرة.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

engineering AGI and theories Moussaif - وحدة 364

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية محكمة الهدم التوليدية / Generative Court Demolition Theory. النقد لا يقتل النموذج؛ يحول العيب إلى تعريف أو أداة أو محور بحث أو تخفيض رتبة معرفية.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

Moussaif engineering AGI and theories - وحدة 365

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الانتروبيا الاختراعية المضبوطة / Governed Inventive Entropy Theory. الابتكار الأعلى يأتي من رفع مساحة البدائل ثم كبحها بمراتب الدليل والاختبار والأثر العالمي.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

Moussaif engineering AGI and theories - وحدة 366

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الحاكمية الوعائية متعددة النماذج / Multi-Model Vascular Governance Theory. لا تكون القدرة في نموذج واحد، بل في طريقة توزيع الانتباه والحساب والذاكرة والمحكمة بين أوعية متخصصة.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

Moussaif engineering AGI and theories - وحدة 367

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية المهارة الفائقة كبرنامج تشغيل معرفي / Hyperskill-as-Cognitive-Operating-Program. المهارة الفائقة ليست tpmorp؛ إنها trigger + state + tools + verifier + rule update + retirement. condition

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

Moussaif engineering AGI and theories - وحدة 368

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الحلقة الفائقة المصعدة / Hyperloop Ascending Theory. كل حلقة تفكير يجب أن تعود بنتيجة تغير قاعدة المعرفة أو طريقة التوجيه، لا بمجرد إجابة عابرة.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

Moussaif engineering AGI and theories - وحدة 369

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية محكمة الهدم التوليدية / Generative Demolition Court Theory. النقد لا يقتل النموذج؛ يحول العيب إلى تعريف أو أداة أو محور بحث أو تخفيض رتبة معرفية.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

Moussaif engineering AGI and theories - وحدة 370

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الانتروبيا الاختراعية المضبوطة / Governed Inventive Entropy Theory. الابتكار الأعلى يأتي من رفع مساحة البدائل ثم كبحها بمراتب الدليل والاختبار والأثر العالمي.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

Moussaif engineering AGI and theories - وحدة 371

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الحاكمية الوعائية متعددة النماذج / Multi-Model Vascular Governance Theory. لا تكون القدرة في نموذج واحد، بل في طريقة توزيع الانتباه والحساب والذاكرة والمحكمة بين أوعية متخصصة.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة LLM systems عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

engineering AGI and theories Moussaif - وحدة 372

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية المهارة الفائقة كبرنامج تشغيل معرفي / .Hyperskill-as-Cognitive-Operating-Program المهارة الفائقة ليست tpmorp؛ إنها trigger + state + tools + verifier + rule update + retirement. condition

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار .routing
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

Moussaif engineering AGI and theories - وحدة 373

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الحلقة الفائقة المصعدة / Hyperloop Ascending Theory. كل حلقة تفكير يجب أن تعود بنتيجة تغير قاعدة المعرفة أو طريقة التوجيه، لا بمجرد إجابة عابرة.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

engineering AGI and theories Moussaif - وحدة 374

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية محكمة الهدم التوليدية / Generative Court Demolition Theory. النقد لا يقتل النموذج؛ يحول العيب إلى تعريف أو أداة أو محور بحث أو تخفيض رتبة معرفية.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

Moussaif engineering AGI and theories - وحدة 375

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الانتروبيا الاختراعية المضبوطة / Governed Inventive Entropy Theory. الابتكار الأعلى يأتي من رفع مساحة البدائل ثم كبحها بمراتب الدليل والاختبار والأثر العالمي.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

Moussaif engineering AGI and theories - وحدة 376

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الحاكمية الوعائية متعددة النماذج / Multi-Model Vascular Governance Theory. لا تكون القدرة في نموذج واحد، بل في طريقة توزيع الانتباه والحساب والذاكرة والمحكمة بين أوعية متخصصة.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

engineering AGI and theories Moussaif - وحدة 377

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية المهارة الفائقة كبرنامج تشغيل معرفي / Hyperskill-as-Cognitive-Operating-Program. المهارة الفائقة ليست tpmorp؛ إنها trigger + state + tools + verifier + rule update + retirement. condition

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

Moussaif engineering AGI and theories - وحدة 378

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الحلقة الفائقة المصعدة / Hyperloop Ascending Theory. كل حلقة تفكير يجب أن تعود بنتيجة تغير قاعدة المعرفة أو طريقة التوجيه، لا بمجرد إجابة عابرة.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

Moussaif engineering AGI and theories - وحدة 379

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية محكمة الهدم التوليدية / Generative Court Demolition Theory. النقد لا يقتل النموذج؛ يحول العيب إلى تعريف أو أداة أو محور بحث أو تخفيض رتبة معرفية.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

Moussaif engineering AGI and theories - وحدة 380

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الكبرى: نظرية الانتروبيا الاختراعية المضبوطة / Entropy Inventive Governed Theory. الابتكار الأعلى يأتي من رفع مساحة البدائل ثم كبحها بمراتب الدليل والاختبار والأثر العالمي.

من أي نقص خرجت؟	من نقص تحويل تعدد النماذج إلى حاكمية قابلة للاختبار.
ما المشكلة التي تحلها؟	فوضى stnega، تضخم الادعاء، وغياب معيار routing.
كيف تختبر؟	بمقارنة integrity. world-effect and fidelity, evidence repair, error latency,
حدها	لا تثبت IGA؛ تثبت تحسنا تشغيليا قابلا للقياس.

النسخة المعدلة بعد محكمة الهدم: هذه النظرية ليست ادعاء وصول إلى IGA، بل برنامج هندسي لرفع قدرة systems LLM عبر توزيع الحساب والذاكرة والمحكمة.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

engineering AGI and theories Moussaif - وحدة 381

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: سلم مراتب القدرة. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "ةردقلا بتارم ملس"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

engineering AGI and theories Moussaif - وحدة 382

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: عتبة الانتقال من loop إلى hyperskill. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "عتبة الانتقال من loop إلى hyperskill"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

383 وحدة - engineering AGI and theories Moussaif

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: ذاكرة الأسباب لا ذاكرة النصوص. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "ذاكرة الأسباب لا ذاكرة النصوص"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

engineering AGI and theories Moussaif - وحدة 384

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: توجيه الحساب بالفرق الفاعل. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "لعافلا قرفلاب باسحلا هيجوت"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

Moussaif engineering AGI and theories - وحدة 385

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: تعقيد التعقيد. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "تعقيد التعقيد"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

إختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

Moussaif engineering AGI and theories - وحدة 386

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: الزمن الداخلي للوكالة. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "الزمن الداخلي للوكالة"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 387 - engineering AGI and theories Moussaif

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: الاختطاف الوعائي. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "الاختطاف الوعائي"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

engineering AGI and theories Moussaif - وحدة 388

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: التصحيح بالمسار المنافس. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "التصحيح بالمسار المنافس"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

engineering AGI and theories Moussaif - وحدة 389

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: البرهان التشغيلي. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "البرهان التشغيلي"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

engineering AGI and theories Moussaif - وحدة 390

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: التحقق متعدد الأوعية. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "التحقق متعدد الأوعية"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

engineering AGI and theories Moussaif - وحدة 391

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: GraphRAG الحاكمي. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "GraphRAG الحاكمي"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

engineering AGI and theories Moussaif - وحدة 392

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: الاستقلال السيادي للنموذج. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "الاستقلال السيادي للنموذج"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

engineering AGI and theories Moussaif - وحدة 393

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: سلم مراتب القدرة. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "ةردقلا بتارم ملس"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

Moussaif engineering AGI and theories - وحدة 394

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: عتبة الانتقال من loop إلى hyperskill. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "عتبة الانتقال من loop إلى hyperskill"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

engineering AGI and theories Moussaif - وحدة 395

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: ذاكرة الأسباب لا ذاكرة النصوص. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "ذاكرة الأسباب لا ذاكرة النصوص"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

engineering AGI and theories Moussaif - وحدة 396

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: توجيه الحساب بالفرق الفاعل. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "لعافلا قرفلاب باسحلا هيجوت"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

engineering AGI and theories Moussaif - وحدة 397

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: تعقيد التعقيد. وظيفتها ضبط مفهوم جزئي داخل نظام .Hermes كل نظرية فرعية يجب أن تتحول إلى test أو rule أو .schema

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "تعقيد التعقيد"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback .rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

إختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition .retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons .of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict, .patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 398 - engineering AGI and theories Moussaif

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: الزمن الداخلي للوكالة. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "الزمن الداخلي للوكالة"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

engineering AGI and theories Moussaif - وحدة 399

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: الاختطاف الوعائي. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "الاختطاف الوعائي"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

engineering AGI and theories Moussaif - وحدة 400

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: التصحيح بالمسار المنافس. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "التصحيح بالمسار المنافس"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

engineering AGI and theories Moussaif - وحدة 401

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: البرهان التشغيلي. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "البرهان التشغيلي"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

engineering AGI and theories Moussaif - وحدة 402

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: التحقق متعدد الأوعية. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "التحقق متعدد الأوعية"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

engineering AGI and theories Moussaif - وحدة 403

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: GraphRAG الحاكمي. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "GraphRAG الحاكمي"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

Moussaif engineering AGI and theories - وحدة 404

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: الاستقلال السيادي للنموذج. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "الاستقلال السيادي للنموذج"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

engineering AGI and theories Moussaif - وحدة 405

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: سلم مراتب القدرة. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "ةردقلا بتارم ملس"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 406 - engineering AGI and theories Moussaif

Version 2: integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: عتبة الانتقال من loop إلى hyperskill. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "عتبة الانتقال من loop إلى hyperskill"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

engineering AGI and theories Moussaif - وحدة 407

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: ذاكرة الأسباب لا ذاكرة النصوص. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "ذاكرة الأسباب لا ذاكرة النصوص"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rule. rollback. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

engineering AGI and theories Moussaif - وحدة 408

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: توجيه الحساب بالفرق الفاعل. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "لعاقلا قرفلاب باسحلا هيجوت"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

engineering AGI and theories Moussaif - وحدة 409

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: تعقيد التعقيد. وظيفتها ضبط مفهوم جزئي داخل نظام .Hermes كل نظرية فرعية يجب أن تتحول إلى test أو rule أو .schema

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "تعقيد التعقيد"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

إختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك retirement .condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement .condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

engineering AGI and theories Moussaif - وحدة 410

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

النظرية الفرعية: الزمن الداخلي للوكالة. وظيفتها ضبط مفهوم جزئي داخل نظام Hermes. كل نظرية فرعية يجب أن تتحول إلى test أو rule أو schema.

- تضبط مفهوما واحدا بدلا من تضخيمه.
- تربط المفهوم بأداة أو مقياس.
- تملك حد فشل واضحا.
- تدخل في patch memory عند نجاحها.

```
sub_theory.name = "الزمن الداخلي للوكالة"
sub_theory.test = "ablation + adversarial cases + routing trace"
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons of Memory.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

protocols metrics, Tools, - وحدة 411

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: MCD-Fusion ،Gate جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في Hermes OS، Agent

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons، of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch، verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons، of Memory

protocols metrics, Tools, - وحدة 412

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Router. Compute Vascular جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

protocols metrics, Tools, - وحدة 413

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Hyperskill. Compiler جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في Hermes OS. Agent

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

protocols metrics, Tools, - وحدة 414

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Court. Demolition Recursive جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

protocols metrics, Tools, - وحدة 415

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Governor. Entropy Inventive جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

protocols metrics, Tools, - وحدة 416

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: World-Effect ،Loop جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في Hermes OS، Agent

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ،ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

protocols metrics, Tools, - وحدة 417

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Memory. Self-Reconstruction جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

protocols metrics, Tools, - وحدة 418

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Sovereignty Model .Matrix جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition .retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons .of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch .verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition .retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

protocols metrics, Tools, - وحدة 419

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Fugu-Style Planner. Scaffold جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

protocols metrics, Tools, - وحدة 420

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Veracity Fusion. Ledger جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

protocols metrics, Tools, - وحدة 421

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Agentic Failure Taxonomy. جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في Agent Hermes OS.

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

protocols metrics, Tools, - وحدة 422

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Binder. Capability-Role جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في Hermes OS. Agent

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

protocols metrics, Tools, - وحدة 423

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Abstraction. Provider OpenRouter-NIM جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

protocols metrics, Tools, - وحدة 424

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Expander. Causal GraphRAG جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

protocols metrics, Tools, - وحدة 425

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Converter. Prompt-to-Protocol جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

protocols metrics, Tools, - وحدة 426

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Generator، Path Red-Team جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS، Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory، patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

protocols metrics, Tools, - وحدة 427

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Annotator. Rank Evidence جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

protocols metrics, Tools, - وحدة 428

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Dial. Latency-Quality جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في Hermes OS. Agent

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

protocols metrics, Tools, - وحدة 429

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Executor، Sandboxed Tool-Use جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS، Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons، of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

protocols metrics, Tools, - وحدة 430

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Detector. Trap Consensus جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

protocols metrics, Tools, - وحدة 431

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Retirement Skill. Rule جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في Hermes OS. Agent

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

protocols metrics, Tools, - وحدة 432

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Simulation Benchmark ،Guard جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS ،Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback ،rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

protocols metrics, Tools, - وحدة 433

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Compression Knowledge .Furnace جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS .Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch .verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement .condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons .of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

protocols metrics, Tools, - وحدة 434

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Governor. Cost Multi-Agent جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

protocols metrics, Tools, - وحدة 435

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Hermes Final Answer Forge. جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

protocols metrics, Tools, - وحدة 436

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: MCD-Fusion Gate. جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في Hermes OS. Agent

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

protocols metrics, Tools, - وحدة 437

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Router، Compute Vascular جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS، Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback، rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

protocols metrics, Tools, - وحدة 438

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Hyperskill. Compiler جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في Hermes OS. Agent

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

protocols metrics, Tools, - وحدة 439

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Court. Demolition Recursive جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

protocols metrics, Tools, - وحدة 440

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Governor. Entropy Inventive جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

protocols metrics, Tools, - وحدة 441

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: World-Effect ،Loop جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في Hermes OS ،Agent

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

protocols metrics, Tools, - وحدة 442

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Memory. Self-Reconstruction جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

protocols metrics, Tools, - وحدة 443

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Matrix. Sovereignty Model جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

protocols metrics, Tools, - وحدة 444

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Planner. Scaffold Fugu-Style جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

protocols metrics, Tools, - وحدة 445

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

الأداة: Ledger. Veracity Fusion جاءت لعلاج نقص تعريفي أو سببي أو قياسي أو تطبيقي. لذلك لا تضاف كزينة، بل كجزء تشغيل في OS. Agent Hermes

input	memory candidates, model constraints, context, task,
steps	update -> fuse -> attack -> run -> route -> classify
output	patch rule ledger, evidence answer, decision,
misuse	استعمالها كاسم قوي بلا schema أو test

الأداة الجيدة لا تقنع باسمها؛ تقنع بقدرتها على تغيير سلوك النظام عند الاختبار. إن لم تغير routing أو verification أو memory فهي مصطلح زائد يجب خفضه أو حذفه.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

protocols metrics, Tools, - وحدة 446

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: Meta-Complexity / MCD يستعمل في Differentiation. dashboard. benchmark promotion, retirement, routing, لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
MCD = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

protocols metrics, Tools, - وحدة 447

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: Vascular Governance .Index يستعمل في ,promotion benchmark .dashboard لا يقبل إلا إذا ارتبط بecart وقابلية .ablation

range	ordinal ranked or 0.0 - 1.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
VGI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback .rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition .retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

protocols metrics, Tools, - وحدة 448

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: Hyperskill / HSI Stability Index. يستعمل في promotion, retirement, routing, benchmark dashboard. لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 0.0 - 1.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
HSI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

protocols metrics, Tools, - وحدة 449

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: FEQ / Evidence Fusion Quality. يستعمل في dashboard. benchmark promotion, retirement, routing, لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
FEQ = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

protocols metrics, Tools, - وحدة 450

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: Recursive Correction / RCR. Rate يستعمل في dashboard. benchmark promotion, retirement, routing, لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 0.0 - 1.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
RCR = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

protocols metrics, Tools, - وحدة 451

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: SRI / Resilience Sovereignty Index. يستعمل في promotion, retirement, routing, benchmark. dashboard لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
SRI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

protocols metrics, Tools, - وحدة 452

expansion: integrated 2 Version حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: World-Effect / WEI يستعمل في Integrity. benchmark promotion, retirement, routing, dashboard. لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
WEI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

protocols metrics, Tools, - وحدة 453

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: Meta-Complexity / MCD Differentiation. يستعمل في dashboard. benchmark promotion, retirement, routing, لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 0.0 - 1.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
MCD = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

protocols metrics, Tools, - وحدة 454

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: Vascular Governance ،Index يستعمل في ,promotion benchmark ،dashboard لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 0.0 - 1.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
VGI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: ,task ,model ,role ,evidence ,verdict .patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك .condition retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: ,task ,model ,role ,evidence ,verdict .patch من دون traces لا يوجد تعلم ولا مقارنة.

protocols metrics, Tools, - وحدة 455

expansion: integrated 2 Version حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: HSI / Hyperskill Stability Index. يستعمل في dashboard. benchmark promotion, retirement, routing, لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 0.0 - 1.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
HSI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

protocols metrics, Tools, - وحدة 456

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: FEQ / Evidence Fusion ،Quality يستعمل في ,promotion benchmark ،dashboard لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
FEQ = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

protocols metrics, Tools, - وحدة 457

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: Recursive Correction Rate / RCR. يستعمل في promotion, retirement, routing, benchmark. dashboard لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 0.0 - 1.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
RCR = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

protocols metrics, Tools, - وحدة 458

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: SRI / Sovereignty Resilience Index. يستعمل في promotion, retirement, routing, benchmark. dashboard لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
SRI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

protocols metrics, Tools, - وحدة 459

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: WEI / World-Effect Integrity. يستعمل في dashboard. benchmark promotion, retirement, routing, لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
WEI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

protocols metrics, Tools, - وحدة 460

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: MCD / Meta-Complexity Differentiation. يستعمل في dashboard. benchmark promotion, retirement, routing, لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
MCD = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

protocols metrics, Tools, - وحدة 461

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: VGI / Vascular Governance Index. يستعمل في dashboard. benchmark promotion, retirement, routing, لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
VGI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

protocols metrics, Tools, - وحدة 462

expansion: integrated 2 Version حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: HSI / Hyperskill Stability Index. يستعمل في benchmark promotion, retirement, routing, dashboard. لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
HSI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

protocols metrics, Tools, - وحدة 463

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: FEQ / Evidence Fusion Quality. يستعمل في dashboard. benchmark promotion, retirement, routing, لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 0.0 - 1.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
FEQ = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

protocols metrics, Tools, - وحدة 464

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: Recursive Correction / RCR. Rate يستعمل في benchmark promotion, retirement, routing, dashboard. لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
RCR = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

protocols metrics, Tools, - وحدة 465

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: SRI / Sovereignty Resilience Index. يستعمل في promotion, retirement, routing, benchmark dashboard. لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 0.0 - 1.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
SRI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. لا يوجد تعلم ولا مقارنة من دون traces.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. لا يوجد تعلم ولا مقارنة من دون traces.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

protocols metrics, Tools, - وحدة 466

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: WEI / World-Effect Integrity. يستعمل في promotion, retirement, routing, benchmark. dashboard لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
WEI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

protocols metrics, Tools, - وحدة 467

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: Meta-Complexity / MCD. Differentiation يستعمل في dashboard. benchmark promotion, retirement, routing, لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
MCD = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rule. rollback هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

protocols metrics, Tools, - وحدة 468

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: VGI / Vascular Governance Index. يستعمل في dashboard. benchmark promotion, retirement, routing, لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
VGI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

protocols metrics, Tools, - وحدة 469

expansion: integrated 2 Version حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: HSI / Hyperskill Stability Index. يستعمل في dashboard. benchmark promotion, retirement, routing, لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
HSI = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

protocols metrics, Tools, - وحدة 470

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المقياس: FEQ / Evidence Fusion ،Quality يستعمل في ,benchmark promotion, retirement, routing ،dashboard لا يقبل إلا إذا ارتبط بecart وقابلية ablation.

range	ordinal ranked or 1.0 - 0.0
evidence minimum	sample review human + result verifier + trace
condition failure	score لا يتحسن عبر ablation أو يتضخم بلا أثر
use	dashboard benchmark promotion, retirement, routing,

```
FEQ = normalize(weighted_sum(features) - risk_penalty)
```

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: ,task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 471 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: SWE-Bench style. repair هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

وحدة 472 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Terminal task. execution هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 473 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Live coding. هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 474 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: GPQA-style science reasoning. هذا الباب لا يقدم نتائج حقيقية؛ يقدم design targets وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن داخلي v2: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 475 - design benchmark and Evaluation

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: broad HLE-style ،difficulty هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation ،label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict ،patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict ،patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 476 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Long-context ،synthesis. هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label ،simulation.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement ،condition. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 477 - design benchmark and Evaluation

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Multi-hop ،GraphRAG هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label ،simulation.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback ،rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 478 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Tool-use. reliability هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

design benchmark and Evaluation - وحدة 479

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Agent، planning هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation، label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble، ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble، ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 480 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Security ،review هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label ،simulation.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 481 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Citation ،fidelity. هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation ،label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement ،condition. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

وحدة 482 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Latency-cost. tradeoff هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 483 - design benchmark and Evaluation

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Self-reconstruction. memory هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 484 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: SWE-Bench style. repair هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن داخلي v2: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 485 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Terminal task. execution هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label. simulation

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 486 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Live coding. هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 487 - design benchmark and Evaluation

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: GPQA-style science reasoning. هذا الباب لا يقدم نتائج حقيقية؛ يقدم design targets وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 488 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: broad HLE-style، difficulty. هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation، label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

وحدة 489 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Long-context. synthesis هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label. simulation

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 490 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Multi-hop ،GraphRAG هذا الباب لا يقدم نتائج حقيقية، يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation ،label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 491 - design benchmark and Evaluation

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Tool-use. reliability هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

وحدة 492 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Agent ،planning هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation ،label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback ،rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement ،condition. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ،ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 493 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Security ،review هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label ،simulation.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement، condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 494 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Citation ،fidelity. هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation ،label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل ،prompt/model/provider/time
- افصل evaluation عن ،marketing

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو ،prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من ،traces كل trace هو عينة علمية: ,task ,model ,role ,evidence ,verdict .patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك ،retirement ،condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ،ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من ،traces كل trace هو عينة علمية: ,task ,model ,role ,evidence ,verdict .patch من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 495 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Latency-cost. tradeoff هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 496 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Self-reconstruction. memory هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 497 - design benchmark and Evaluation

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: SWE-Bench ،style ،repair هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label ،simulation.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback ،rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 498 - design benchmark and Evaluation

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Terminal task. execution هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

وحدة 499 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Live coding. هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 500 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: GPQA-style science reasoning. هذا الباب لا يقدم نتائج حقيقية؛ يقدم design targets وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 501 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: broad HLE-style، difficulty. هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation، label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

وحدة 502 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Long-context ،synthesis. هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label ،simulation.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback ،rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement ،condition. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 503 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Multi-hop ،GraphRAG هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation ،label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 504 - design benchmark and Evaluation

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Tool-use ،reliability. هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label ،simulation.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 505 - design benchmark and Evaluation

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Agent ،planning هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation ،label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل ،prompt/model/provider/time
- افصل evaluation عن ،marketing

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو ،prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من ،traces كل trace هو عينة علمية: task, model, role, evidence, verdict ،patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ،ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من ،traces كل trace هو عينة علمية: task, model, role, evidence, verdict ،patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 506 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Security ،review هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label ،simulation.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement ،condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 507 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Citation ،fidelity. هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation ،label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback ،rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 508 - design benchmark and Evaluation

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Latency-cost. tradeoff هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

وحدة 509 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Self-reconstruction. memory هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label. simulation

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 510 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: SWE-Bench style. repair هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label. simulation

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 511 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Terminal task. execution هذا الباب لا يقدم نتائج حقيقية؛ يقدم design targets وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

وحدة 512 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Live coding. هذا الباب لا يقدم نتائج حقيقية، يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation. label

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 513 - design benchmark and Evaluation

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: GPQA-style science reasoning. هذا الباب لا يقدم نتائج حقيقية؛ يقدم design targets وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 514 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: broad HLE-style ،difficulty هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label ،simulation.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 515 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Long-context. synthesis هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو label. simulation

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 516 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Multi-hop ،GraphRAG هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation ،label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement ،condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 517 - design benchmark and Evaluation

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Tool-use ،reliability. هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation ،label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback ،rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 518 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Agent ،planning هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation ،label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

وحدة 519 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Security ،review هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation ،label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition ،retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 520 - design benchmark and Evaluation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور التقييم: Citation ،fidelity. هذا الباب لا يقدم نتائج حقيقية؛ يقدم targets design وبروتوكولات محاكاة يجب أن تنفذ لاحقا. أي رقم بلا تشغيل فعلي يبقى فرضية أو simulation ،label.

dimension	design test
baseline	call direct model best single
fusion	mode multi-agent routed Hermes
metric	risk cost, latency, fidelity, citation rate, repair accuracy,
ablation	fallback remove / GraphRAG remove / memory remove / court remove

- لا تقارن نماذج مختلفة دون نفس المدخلات.
- لا تعتبر vote majority دليلا نهائيا.
- سجل prompt/model/provider/time.
- افصل evaluation عن marketing.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch، verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

courts demolition and program Research - وحدة 521

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(theory. routing السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في theory routing
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

courts demolition and program Research - وحدة 522

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(learned ،scaffolds السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في scaffolds learned
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

courts demolition and program Research - وحدة 523

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(compiler ،hyperskill السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في compiler hyperskill
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons .of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

courts demolition and program Research - وحدة 524

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(governance ،GraphRAG السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في governance GraphRAG
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement .condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble .ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

courts demolition and program Research - وحدة 525

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(courts. safety السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في courts safety
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

courts demolition and program Research - وحدة 526

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(sovereignty. model السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في sovereignty model
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

courts demolition and program Research - وحدة 527

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(reasons. of memory السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في reasons of memory
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

courts demolition and program Research - وحدة 528

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(sandbox، tool-use السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في sandbox tool-use
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch، verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

courts demolition and program Research - وحدة 529

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(benchmarks .agentic السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في benchmarks agentic
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

courts demolition and program Research - وحدة 530

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(engineering، KB AGI Arabic السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في engineering KB AGI Arabic
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

courts demolition and program Research - وحدة 531

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(theory. routing السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في theory routing
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

courts demolition and program Research - وحدة 532

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(scaffolds. learned السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في scaffolds learned
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

courts demolition and program Research - وحدة 533

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(compiler ،hyperskill السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في compiler hyperskill
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch .verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition .retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons .of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

courts demolition and program Research - وحدة 534

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(governance ،GraphRAG السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في governance GraphRAG
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: ,patch .verdict, evidence, role, model, task من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition .retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: ,patch .verdict, evidence, role, model, task من دون traces لا يوجد تعلم ولا مقارنة.

courts demolition and program Research - وحدة 535

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(courts. safety السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في courts safety
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

courts demolition and program Research - وحدة 536

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(sovereignty. model السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في sovereignty model
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

courts demolition and program Research - وحدة 537

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(reasons. of memory السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في reasons of memory
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rule. rollback هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

courts demolition and program Research - وحدة 538

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(sandbox، tool-use السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في sandbox tool-use
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement، condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch، verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement، condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

courts demolition and program Research - وحدة 539

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)سنة:(benchmarks. agentic السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في benchmarks agentic
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

courts demolition and program Research - وحدة 540

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(engineering، KB AGI Arabic السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في engineering KB AGI Arabic
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement، condition. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

courts demolition and program Research - وحدة 541

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(theory، routing السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في theory routing
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

courts demolition and program Research - وحدة 542

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(scaffolds، learned السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في scaffolds learned
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

courts demolition and program Research - وحدة 543

expansion: integrated 2 Version حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(compiler. hyperskill السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في compiler hyperskill
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

courts demolition and program Research - وحدة 544

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(governance ،GraphRAG السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في governance GraphRAG
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict ،patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict ،patch من دون traces لا يوجد تعلم ولا مقارنة.

courts demolition and program Research - وحدة 545

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(courts، safety السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في courts safety
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement، condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

courts demolition and program Research - وحدة 546

Version 2 :integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(sovereignty. model السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في sovereignty model
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

courts demolition and program Research - وحدة 547

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(reasons of memory. السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في reasons of memory
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rule. rollback هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

courts demolition and program Research - وحدة 548

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(sandbox، tool-use السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في sandbox tool-use
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement، condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch، verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement، condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

courts demolition and program Research - وحدة 549

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(benchmarks. agentic السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في benchmarks agentic
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

courts demolition and program Research - وحدة 550

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(engineering، KB AGI Arabic السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في engineering KB AGI Arabic
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement، condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

courts demolition and program Research - وحدة 551

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(theory، routing السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في theory routing
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

courts demolition and program Research - وحدة 552

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(scaffolds، learned السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في scaffolds learned
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

courts demolition and program Research - وحدة 553

expansion: integrated 2 Version حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(compiler. hyperskill السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في compiler hyperskill
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

courts demolition and program Research - وحدة 554

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)ثلاث سنوات:(governance ،GraphRAG السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في governance GraphRAG
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict ،patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement ،condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict ،patch من دون traces لا يوجد تعلم ولا مقارنة.

courts demolition and program Research - وحدة 555

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(courts. safety السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في courts safety
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition، المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

courts demolition and program Research - وحدة 556

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(model .sovereignty السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في sovereignty model
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition .retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

courts demolition and program Research - وحدة 557

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(reasons. of memory السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في reasons of memory
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rule. rollback هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

courts demolition and program Research - وحدة 558

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(sandbox، tool-use السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في sandbox tool-use
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble، يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons، of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces، كل trace هو عينة علمية: patch، verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

courts demolition and program Research - وحدة 559

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(benchmarks ،agentic السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في benchmarks agentic
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

courts demolition and program Research - وحدة 560

Version integrated 2 :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(engineering، KB AGI Arabic السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في engineering KB AGI Arabic
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement، condition. المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

courts demolition and program Research - وحدة 561

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(theory. routing السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في theory routing
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

courts demolition and program Research - وحدة 562

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(scaffolds .learned السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في scaffolds learned
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback .rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons .of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch .verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition .retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons .of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

courts demolition and program Research - وحدة 563

expansion: integrated 2 Version حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(compiler ،hyperskill السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في compiler hyperskill
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict ،patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement ،condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

courts demolition and program Research - وحدة 564

Version integrated 2 :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(governance ،GraphRAG السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في governance GraphRAG
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role <- slice task <- evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict ،patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict ،patch من دون traces لا يوجد تعلم ولا مقارنة.

courts demolition and program Research - وحدة 565

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(courts. safety السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في courts safety
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement .condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

courts demolition and program Research - وحدة 566

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(model .sovereignty السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في sovereignty model
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory .patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition .retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

courts demolition and program Research - وحدة 567

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(reasons. of memory السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في reasons of memory
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rule. rollback هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

courts demolition and program Research - وحدة 568

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(sandbox، tool-use السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في sandbox tool-use
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble، يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch، verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

courts demolition and program Research - وحدة 569

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(benchmarks ،agentic السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في benchmarks agentic
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory ،patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition ،retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

courts demolition and program Research - وحدة 570

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

محور بحثي)عشر سنوات:(engineering، KB AGI Arabic السؤال المركزي: كيف نحول هذا المحور من فكرة جذابة إلى اختبار أو أداة أو ورقة قابلة للنقد؟

problem	غياب معيار صارم في engineering KB AGI Arabic
hypothesis	التوجيه الوعائي مع المحكمة يحسن الجودة مع ضبط التكلفة
method	tracing source + audit human + suite adversarial + ablation
data	patches memory logs, verifier outputs, traces, tasks,
value	تحويل Hermes من prototype إلى program research

محكمة الهدم: قد تكون الجدة هنا لغوية لا جوهرية. لذلك يجب مقارنة كل مفهوم بنظائره في Fugu and GraphRAG, RAG, Reflexion, ToT, ReAct, قبل ادعاء الخرق.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role <- slice task <- evidence <- risk <- output <- verifier <- memory، patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

blueprints Implementation - وحدة 571

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: registry.json. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

blueprints Implementation - وحدة 572

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: router.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

blueprints Implementation - وحدة 573

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: fusion.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 574 - blueprints Implementation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: court.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

blueprints Implementation - وحدة 575

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: memory.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

blueprints Implementation - وحدة 576

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: evaluator.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

blueprints Implementation - وحدة 577

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: provider/openrouter.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt. soup

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

blueprints Implementation - وحدة 578

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: provider/nim.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

blueprints Implementation - وحدة 579

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: workflow/code_fusion.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

blueprints Implementation - وحدة 580

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: workflow/deep_research.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 581 - blueprints Implementation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: registry.json. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

وحدة 582 - blueprints Implementation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: router.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

blueprints Implementation - وحدة 583

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: fusion.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

blueprints Implementation - وحدة 584

Version 2 integrated expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: court.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

وحدة 585 - blueprints Implementation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: memory.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

blueprints Implementation - وحدة 586

Version 2 integrated: expansion :حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: evaluator.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

blueprints Implementation - وحدة 587

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: provider/openrouter.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt. soup

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

blueprints Implementation - وحدة 588

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: provider/nim.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

blueprints Implementation - وحدة 589

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: workflow/code_fusion.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

blueprints Implementation - وحدة 590

Version 2 integrated: expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: workflow/deep_research.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

blueprints Implementation - وحدة 591

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: registry.json. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

blueprints Implementation - وحدة 592

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: router.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

blueprints Implementation - وحدة 593

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: fusion.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

blueprints Implementation - وحدة 594

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: court.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

blueprints Implementation - وحدة 595

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: memory.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

blueprints Implementation - وحدة 596

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: evaluator.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

blueprints Implementation - وحدة 597

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: provider/openrouter.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

blueprints Implementation - وحدة 598

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: provider/nim.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

blueprints Implementation - وحدة 599

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: workflow/code_fusion.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

blueprints Implementation - وحدة 600

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: workflow/deep_research.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

blueprints Implementation - وحدة 601

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: registry.json. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

blueprints Implementation - وحدة 602

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: router.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط، يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

blueprints Implementation - وحدة 603

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: fusion.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

blueprints Implementation - وحدة 604

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: court.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

blueprints Implementation - وحدة 605

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: memory.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

blueprints Implementation - وحدة 606

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: evaluator.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى soup. prompt

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, verdict, evidence, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

blueprints Implementation - وحدة 607

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: provider/openrouter.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt. soup

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 608 - blueprints Implementation

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: provider/nim.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

وحدة 609 - blueprints Implementation

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: workflow/code_fusion.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

blueprints Implementation - وحدة 610

Version 2 integrated: expansion: حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

ملف التنفيذ: workflow/deep_research.ts. يجب أن يكون كل ملف في Hermes قابلا للاختبار منفصلا، لأن orchestration agentic إذا لم يفصل layers يتحول إلى prompt soup.

module	responsibility
router	select + score + classify
worker	output schema + prompt role
court	rank + repair + attack
fusion	only claims supported merge
memory	patches and reasons store

```
type Trace = {task_id:string, model:string, role:string, input_hash:string, output_hash:string, verdict:string, patch?:s
```

القاعدة الصارمة: لا جواب نهائي دون trace. لا trace دون role. لا role دون معيار اختيار. لا معيار دون ذاكرة تعلم.

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

وحدة 611 - log build glossary, Sources,

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المصدر: Fugu Sakana Technical Report. المعرف/الرابط: arXiv 2606.21228. موضعه في البناء: orchestration, multi-agent learned scaffolds, adaptive Fugu/Fugu-Ultra split. speed-quality

درجة الاعتماد	مصدر أولي/بحثي أو توثيق رسمي حسب النوع
حدوده	لا يثبت نظرية fiassuoM؛ يثبت فقط خلفية تقنية أو سلوك أداة
طريقة الاستعمال	تعريف، baseline evaluation or guard, implementation comparison,

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قوياً لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

وحدة 612 - log build glossary, Sources,

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المصدر: Fugu Sakana. Release المعرف/الرابط: sakana.ai/fugu-release. موضعه في البناء: single API interface for model, selection, delegation, verification. synthesis,

درجة الاعتماد	مصدر أولي/بحثي أو توثيق رسمي حسب النوع
حدوده	لا يثبت نظرية MuossaiF؛ يثبت فقط خلفية تقنية أو سلوك أداة
طريقة الاستعمال	تعريف، comparison, implementation baseline evaluation or guard,

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict, patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

وحدة 613 - log build glossary, Sources,

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المصدر: OpenRouter Models Free Collection. المعرف/الرابط: openrouter.ai/collections/free-models. موضعه في البناء: free catalog. model warnings, provider/privacy router, free routes,

درجة الاعتماد	مصدر أولي/بحثي أو توثيق رسمي حسب النوع
حدوده	لا يثبت نظرية fiassuoM؛ يثبت فقط خلفية تقنية أو سلوك أداة
طريقة الاستعمال	تعريف، baseline evaluation or guard, implementation comparison,

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد؛ القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory -> patch. إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

وحدة 614 - log build glossary, Sources,

Version 2 integrated :expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المصدر: NVIDIA NIM LLM. API المعرف/الرابط: docs.nvidia.com/nim/large-language-models/latest/api-reference.html.
موضعه في البناء: interface inference OpenAI-compatible. NIM deployable for services

درجة الاعتماد	مصدر أولي/بحثي أو توثيق رسمي حسب النوع
حدوده	لا يثبت نظرية fiassuoM؛ يثبت فقط خلفية تقنية أو سلوك أداة
طريقة الاستعمال	تعريف، baseline evaluation or guard, implementation comparison,

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء verifier داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.
GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.
الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.
كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.
تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.
قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.
قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory
الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.
في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.
البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

log build glossary, Sources, - وحدة 615

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المصدر: .ReAct المعرف/الرابط: arXiv .2210.03629 موضعه في البناء: tools/environments. with acting and reasoning interleaving

درجة الاعتماد	مصدر أولي/بحثي أو توثيق رسمي حسب النوع
حدوده	لا يثبت نظرية fiassuoM؛ يثبت فقط خلفية تقنية أو سلوك أداة
طريقة الاستعمال	تعريف، baseline evaluation or guard, implementation comparison,

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء memory-updater داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: slice task -> role -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: patch. verdict, evidence, role, model, task, من دون traces لا يوجد تعلم ولا مقارنة.
محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

وحدة 616 - log build glossary, Sources,

Version 2 integrated: expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المصدر: Tree of Thoughts، المعرف/الرابط: arXiv ،2305.10601 موضعه في البناء: search over multiple reasoning paths, self-evaluation، and backtracking

درجة الاعتماد	مصدر أولي/بحثي أو توثيق رسمي حسب النوع
حدوده	لا يثبت نظرية MuseumFiasso؛ يثبت فقط خلفية تقنية أو سلوك أداة
طريقة الاستعمال	تعريف، baseline evaluation or guard, implementation comparison,

حقن v2 الداخلي: عدسة التعقيد: لا نعالج task كرسالة واحدة، بل كنظام فرق وقيود ومخاطر وبدائل وذاكرة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
سببي	ما الآلية التي تربط الإدخال بالمخرج؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء planner داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory، patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition، retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة MCD: كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Memory of Reasons.

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict، patch. من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

وحدة 617 - log build glossary, Sources,

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المصدر: .Self-Consistency المعرف/الرابط: arXiv .2203.11171 موضعه في البناء: selecting then paths reasoning multiple sampling answers. consistent

درجة الاعتماد	مصدر أولي/بحثي أو توثيق رسمي حسب النوع
حدوده	لا يثبت نظرية fiassuoM؛ يثبت فقط خلفية تقنية أو سلوك أداة
طريقة الاستعمال	تعريف، baseline evaluation or guard, implementation comparison,

حقن v2 الداخلي: عدسة الهندسة: كل مفهوم يجب أن يتحول إلى schema أو trace أو verifier أو metric أو rollback. rule هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
قياسي	ما المؤشر الذي يكشف التحسن أو الفشل؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء coder داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك retirement. condition المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble. ensemble يجمع احتمالات أو إجابات، أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

وحدة 618 - log build glossary, Sources,

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المصدر: .Reflexion المعرف/الرابط: arXiv .2303.11366. موضعه في البناء: agents. for memory episodic and feedback verbal

درجة الاعتماد	مصدر أولي/بحثي أو توثيق رسمي حسب النوع
حدوده	لا يثبت نظرية fiassuoM؛ يثبت فقط خلفية تقنية أو سلوك أداة
طريقة الاستعمال	تعريف، baseline evaluation or guard, implementation comparison,

حقن v2 الداخلي: عدسة السيادة: القدرة ليست في provider واحد، القدرة في قابلية تبديل النموذج دون انهيار السلوك. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تطبيقي	ما الملف أو البروتوكول الذي يشغله؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء critic داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو .prompts الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس .ensemble ensemble يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط، تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من .traces كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

وحدة 619 - log build glossary, Sources,

Version 2 :integrated expansion حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المصدر: RAG. المعرف/الرابط: arXiv. 2005.11401 موضعه في البناء: for memory non-parametric plus parametric generation، knowledge-intensive

درجة الاعتماد	مصدر أولي/بحثي أو توثيق رسمي حسب النوع
حدوده	لا يثبت نظرية fiassuoM؛ يثبت فقط خلفية تقنية أو سلوك أداة
طريقة الاستعمال	تعريف، baseline evaluation or guard, implementation comparison,

حقن v2 الداخلي: عدسة المحكمة: أقوى إضافة هي التي تنجو من اعتراض متخصص لا من مدح لغوي داخلي. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر؛ فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
حدودي	أين تقف النظرية ولا تتكلم؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء retriever داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في Hermes v2 لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء، والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

وحدة 620 - log build glossary, Sources,

Version expansion: integrated 2 حفظ الأصل، حقن داخلي، أدوات، محاكم، برنامج بحث

المصدر: GraphRAG. المعرف/الرابط: GraphRAG Microsoft + arXiv surveys. موضعه في البناء: retrieval, graph-structured relations. multi-hop communities,

درجة الاعتماد	مصدر أولي/بحثي أو توثيق رسمي حسب النوع
حدوده	لا يثبت نظرية fiassuoM؛ يثبت فقط خلفية تقنية أو سلوك أداة
طريقة الاستعمال	تعريف، baseline evaluation or guard, implementation comparison,

حقن v2 الداخلي: عدسة المعرفة: فرز كل claim إلى مصدر، راجح هندسي، تركيب نظري، فرضية، أو استعارة منتجة. هذا الحقن يمنع أن تتحول قاعدة Hermes إلى أسماء كبيرة بلا أثر، فكل اسم يجب أن يملك مدخلا ومخرجا وحدا وفشلا قابلا للملاحظة.

محور الضبط	سؤال المحكمة	أثره في Hermes
تعريفي	هل للمفهوم حد يمنع سوء الاستعمال؟	يحدد هل نرقي الفكرة إلى hyperskill أو نخفضها إلى فرضية
Routing	هل الاختيار مبني على MCD أم على شهرة النموذج؟	يمنع worship vendor ويحفظ قابلية الاستبدال
Fusion	هل تم دمج أدلة أم مجرد تلخيص؟	يفصل synthesis عن summarization
Memory	هل حفظنا سبب القرار؟	يبني تعلم النظام عبر الزمن

قاعدة تشغيلية: عندما يعمل وعاء synthesizer داخل Hermes يجب أن يخرج JSON أو trace قابلا للمحاكمة، لا فقرة بلاغية. المسار الصحيح: role -> slice task -> evidence -> risk -> output -> verifier -> memory. patch إذا غاب واحد من هذه الحقول صار agent مجرد متكلم آخر.

اعتراض قوي: قد تكون فكرة hyperskills إعادة تسمية للمهارات أو prompts. الجواب المنهجي: لا تثبت الجدة بالاسم، بل بثلاثة اختبارات: قابلية النقل بين النماذج، انخفاض الفشل بعد hctap memory، ووجود rule retirement يمنع تراكم المهارات الميتة.

اختبار مصغر: أزل المحكمة مرة، ثم أزل الذاكرة مرة، ثم أزل GraphRAG مرة، وقارن جودة الجواب والأثر العالمي. إن لم يظهر فرق، فالمكون ليس مركزيا ويجب إعادة بنائه أو تخفيض رتبته.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.

قاعدة المعرفة الحية لا تخزن نتائج فقط؛ تخزن سبب القرار، سبب الخطأ، البديل الذي خسر، والقيد الذي غير الاختيار. هذه هي Reasons. of Memory

الخرق المعرفي المنضبط يطلب جرأة في التوليد وتواضعا في الادعاء. لذلك كل نظرية هنا لها حد، وكل أداة لها سوء استعمال، وكل benchmark له شرط تشغيل.

في v2 Hermes لا يصبح النموذج قويا لأنه كبير، بل لأنه موضوع في الدور الصحيح. النموذج الصغير قد يكون أفضل router أو checker إذا كان موضعه حاكما.

البرنامج البحثي يبدأ من traces. كل trace هو عينة علمية: task, model, role, evidence, verdict. patch من دون traces لا يوجد تعلم ولا مقارنة.

محكمة الهدم لا تسأل هل الجواب جميل، بل هل الآلية صحيحة، هل الدليل كاف، هل هناك خلط مستويات، وهل توجد طريقة اختبار تنجو من الاعتراض.

GraphRAG الحاكمي لا يبني graph للعلاقات الاسمية فقط؛ يضيف علاقات سبب، قيد، أداة، اختبار، فشل، ونسخة مصححة، حتى يخدم القرار لا التزيين.

الانتقائية المساعفية في التوجيه تعني اختيار أقل مسار كاف لا أكثر مسار مبهر. الاقتصاد في agents جزء من الذكاء لا نقص في الطموح.

كل hyperskill يجب أن يملك condition. retirement المهارة التي لا يمكن إيقافها تتحول إلى عادة عمياء؛ والقاعدة التي لا تقبل التعديل تتحول إلى عائق.

تمييز إضافي: orchestration ليس ensemble ensemble. يجمع احتمالات أو إجابات؛ أما orchestration فيبني مسارا: من يخطط، من ينفذ، من يراجع، من يسند، ومن يقرر متى يتوقف.

قاعدة :MCD كلما زاد الفرق الفاعل والقيود والمخاطرة، ارتفع حق النظام في تشغيل أكثر من وعاء. أما الطلب المنخفض المخاطرة فيجب أن يبقى سريع المسار.