MCAI-V2 AGI New Technology — neoAGI

تقنيات حاكمية وعائية جديدة لتجاوز النماذج المفكرة الحالية

إنجاز وتركيب: زهير مساعف — Zouhair Moussaif

كتاب بحثي تقني مكثف في بناء تقنية AGI جديدة: ليست وعدا حتميا، بل برنامج اختراع واختبار يستهدف تجاوز حدود النماذج المفكرة الحالية عبر الحاكمية الوعائية، الانتقاءية المساعفية، HAKIM-MCD، ونظام neoAGI.

الطبعة التجريبية البحثية — 140 صفحة A4 — خط Amiri — اتجاه RTL مع عزل الإنجليزية.

صفحة العنوان

MCAI-V2 AGI New Technology — neoAGI

تقنيات حاكمية وعائية جديدة لتجاوز النماذج المفكرة الحالية

إنجاز وتركيب: زهير مساعف — Zouhair Moussaif

موضوع الكتاب: اختراع تقنيات وفريم ووركات ومقاييس قابلة للتنفيذ لبناء جيل مختلف من الذكاء العام الاصطناعي: جيل لا يقوم على زيادة حجم النموذج فقط، بل على حاكمية شروط المعرفة والفعل والتطور الذاتي.

هذا العمل يدمج: **MICAI / MCAI**، نظرية التراكب الوعائي الحاكمي، الانتقاءية المساعفية، HAKIM-MCD، مقاومة الاختطاف، الذاكرة الوعائية، نماذج العالم الإحالية، ومحكمة الهدم.

تنبيه منهجي: لا حتمية بلا تجربة

عبارة «تقنيات توصل إلى AGI» تُستعمل هنا بمعنى: مسارات تقنية ذات قدرة أعلى من المعماريات الحالية إذا نجحت تجريبيا، لا بمعنى ضمان ميتافيزيقي أو إعلان حل نهائي. الحتمية العلمية لا تُمنح بل تُنتزع عبر تكرار تجارب، falsification ،ablation، ومقارنة ضد نماذج منافسة.

ينقسم الكلام في الكتاب إلى ست مراتب: ثابت مدعوم بمصادر، راجح في البحث الحالي، خلافي، استنتاج منطقي، تركيب مساعفي جديد، وتأمل يحتاج اختبارا. كل تقنية مخترعة تمر عبر ثلاث بوابات: تعريف واضح، آلية تشغيل، ومقياس فشل.

> **قاعدة الكتاب:** AGI لا يُختزل في إجابة صحيحة؛ AGI الأقوى هو نظام يحكم شروط إنتاج الجواب، ويعرف متى يتوقف، وكيف يراجع ذاكرته، وكيف يحمي نفسه من الاختطاف، وكيف يحول الفشل إلى تحسين.

جدول المصادر المركزية: الجزء الأول

#	المصدر	المؤلف/المؤسسة	السنة	وظيفته داخل الكتاب
1	CoALA: Cognitive Architectures for Language Agents	Sumers et al.	2023	يوفر لغة معمارية للوكلاء: ذاكرة، أفعال، واتخاذ قرار؛ نستعمله كحد أدنى لا كنهاية.
2	Bridging Generative Networks with the Common Model of Cognition	West et al.	2024	يربط النماذج التوليدية بالمعماريات المعرفية؛ نستخرج منه الحاجة إلى نظام حاكم فوق وحدات مولدة.
3	Learning to Reason with LLMs / o1	OpenAI	2024	يثبت اتجاه train-time/test-time compute، لكنه لا يحل مشكلة الحاكمية والمصدر والاختطاف.
4	Deep Research System Card	OpenAI	2025	يمثل الوكيل الباحث المتصفح؛ نستعمله لتحديد حدود الوكلاء البحثيين لا لنسخهم.
5	AlphaEvolve	Google DeepMind	2025	نموذج لاكتشاف خوارزميات عبر LLM + مقيمات + تطور؛ نوسعه إلى انتقاءية معرفية حاكمة.
6	Darwin Gödel Machine	Zhang et al.	2025	نقطة مرجعية في التحسين الذاتي المفتوح؛ نضيف إليه دستور حاكمية وسجل أسباب ومناعة ضد الانجراف.
7	Revisiting Test-Time Scaling	Chung et al.	2025	يصنف مسارات TTS: sampling/search/trajectory optimization؛ نضيف معيار التنوع المحكوم لا التنوع الخام.
8	Test-Time Scaling Is Not Effective for Knowledge-Intensive Tasks Yet	Zhao et al.	2025	ينبه إلى أن التفكير الأطول قد يزيد الهلوسة؛ دليل على حاجة الحاكمية الإحالية.
9	Memory in the Age of AI Agents	Hu et al.	2025	ينقل الذاكرة من قسمة قصيرة/طويلة إلى forms/functions/dynamics؛ نستعمله لتأسيس الذاكرة الوعائية.
10	A Survey on the Memory Mechanism of LLM-based Agents	Zhang et al.	2024	يمدنا بخريطة تصميم وتقييم الذاكرة؛ نضيف فوقها الإحالة والمراتب والاختطاف.

المصادر لا تُستعمل لإغلاق التفكير، بل لتحديد السقف الحالي، ثم كشف ما لا تقوله: أغلبها يصف الذاكرة أو الاستدلال أو الأداة أو القياس؛ مشروعنا يسأل عن طبقة الحاكمية التي تضبط علاقة هذه العناصر بعضها ببعض.

جدول المصادر المركزية: الجزء الثاني

#	المصدر	المؤلف/المؤسسة	السنة	وظيفته داخل الكتاب
11	World Models for Embodied AI	Li et al.	2025	يعرض محاور نماذج العالم: الوظيفة والزمن والتمثيل المكاني؛ نضيف إليها الإحالة والغرض والحاكمية.
12	Neuro-Symbolic AI for LLM Reasoning	Yang et al.	2025	يظهر أن الدمج العصبي الرمزي طريق واعد؛ نجعله تحت حاكمية لا مجرد تركيب تقني.
13	OWASP Top 10 for LLM Applications	OWASP	2025	يعطي مخاطر مثل prompt injection وسلسلة التوريد؛ نعيد صياغتها كاختطاف حاكمية.
14	ARC-AGI-2	ARC Prize	2025	معيار يضغط الذكاء السائل؛ نستعمله لاختبار النقل والتجريد لا لحسم AGI.
15	SWE-bench / Verified	SWE-bench team	2024-2025	اختبار برمجي واقعي؛ نضيف له سجل الحاكمية لا pass/fail فقط.
16	BrowseComp	Wei et al.	2025	معيار تصفح وبحث؛ نستعمله لاختبار الوكيل الباحث لا النموذج اللغوي وحده.
17	Toward Efficient Agents	Survey project	2025	ينبه إلى كلفة الذاكرة والأدوات والسياق؛ نضيف محاسبة حاكمية للكلفة.
18	Cognitive Architectures Survey	Kotseruba & Tsotsos	2020	يعطي تاريخا واسعا للمعماريات؛ نستعمله كجذر لا كنسخة جاهزة.

تمييز: معيار المصدر هنا ليس الشهرة فقط؛ بل قدرة المصدر على أن يغير تصميما هندسيا أو مقياسا أو اعتراضا. لذلك يدخل OWASP في صلب AGI لأنه يكشف اختطاف الحاكمية، ويدخل ARC/SWE/BrowseComp لأنها تفضح حدود القياس لا لأنها تقيس كل شيء.

فهرس تفصيلي مركز

1. الباب الأول: ما الذي يجب تجاوزه في نماذج التفكير الحالية؟
حدود التفكير الطويل دون إحالة؛ منطق السلسلة الخفية ومشكلة الثقة؛ الوكيل كملحق لا كعقل حاكم؛ المصدر والذاكرة كطبقات غير محكومة؛ فشل AGI حين يساوي القدرة بالإجابة؛ قانون الاختطاف في النماذج المفكرة؛ الهندسة الحالية بين scaling وworkflow؛ لماذا لا تكفي المعمارية Transformer وحدها؛ الحد الفاصل بين ذكاء الاختبار وذكاء العالم

2. الباب الثاني: مبادئ neoAGI في مشروع MCAI-V2
مبدأ الحاكمية قبل التوليد؛ مبدأ الوعاء قبل السياق؛ مبدأ الإحالة قبل الثقة؛ مبدأ الانتقاءية المساعفية؛ مبدأ الزمن الداخلي متعدد المقاييس؛ مبدأ مقاومة الاختطاف؛ مبدأ الإثبات المحمول معرفيا؛ مبدأ الانقلاب المنهجي ضد المسلمات؛ مبدأ التطوير الذاتي المحكوم

3. الباب الثالث: HAKIM-MCD Kernel
تعريف النواة الحاكمة؛ تمثيل القيود الهرمية؛ مقياس MCD كضغط انتقائي؛ تجميع الأدلة والاعتراضات؛ دالة القرار الحاكم؛ سجل الأسباب لا سجل المخرجات فقط؛ حاكمية على الحاكمية؛ مسارات التفكير المتنافسة؛ تدريب النواة عبر محكمة الهدم

4. الباب الرابع: الذاكرة الوعائية لا الذاكرة المتجهية فقط
لماذا vector memory غير كافية؛ وعاء الواقعة ووعاء المصدر؛ وعاء الفشل ووعاء الاعتراض؛ الذاكرة كجهاز مناعة؛ الاسترجاع بالإحالة لا بالقرب الدلالي فقط؛ نسيان محكوم لا حذف عشوائي؛ ضغط التجربة إلى أدوات؛ ذاكرة المستخدم والغرض؛ اختبار الذاكرة عبر الانقطاعات الزمنية

5. الباب الخامس: نماذج العالم الإحالية
من world model إلى referential world model؛ الحالة والفعل والدليل؛ المحاكاة المضادة للواقع؛ نماذج العالم للروبوتات والويب والبحث؛ كشف الفجوة بين العالم والنص؛ محاسبة القيود الفيزيائية والرمزية؛ تجسيد غير جسدي: الوكيل داخل أدوات؛ تحديث العالم عبر الفشل؛ اختبار العالم بمعضلات انتقال المجال

6. الباب السادس: الانتقاءية المساعفية كمحرك تطور
الانتقاء ليس sampling؛ انتقاء المسارات لا الإجابات؛ الضغط المعرفي كمحرك اكتشاف؛ تطور ذاتي مع دستور؛ أرشيف المسارات الحية؛ الطفرة المقيدة والطفرة الممنوعة؛ أثر AlphaEvolve وDGM وحدودهما؛ انتقاء الفرضيات العلمية؛ انتقاء الأدوات والكود

7. الباب السابع: من test-time compute إلى governed-time compute
لماذا التفكير الأطول قد يخدع؛ تنوع المسارات تحت قيود؛ المقيم لا يكفي بلا مراتب دليل؛ search over thoughts vs search over governed states؛ إيقاف التفكير حين يزيد الخطر؛ جدولة الزمن الداخلي؛ التفكير كميزانية لا كزينة؛ التحقق المرحلي؛ معيار العائد المعرفي لكل token

8. الباب الثامن: الهندسة البرمجية لـneoAGI
تصميم OS معرفي فوق النماذج؛ بروتوكول الأدوات؛ سجل أفعال قابل للتدقيق؛ Sandbox معرفي وتنفيذي؛ Knowledge Graph حاكمي؛ Causal Ledger؛ Verifier Mesh؛ Policy-as-Code للحاكمية؛ واجهات API للنواة

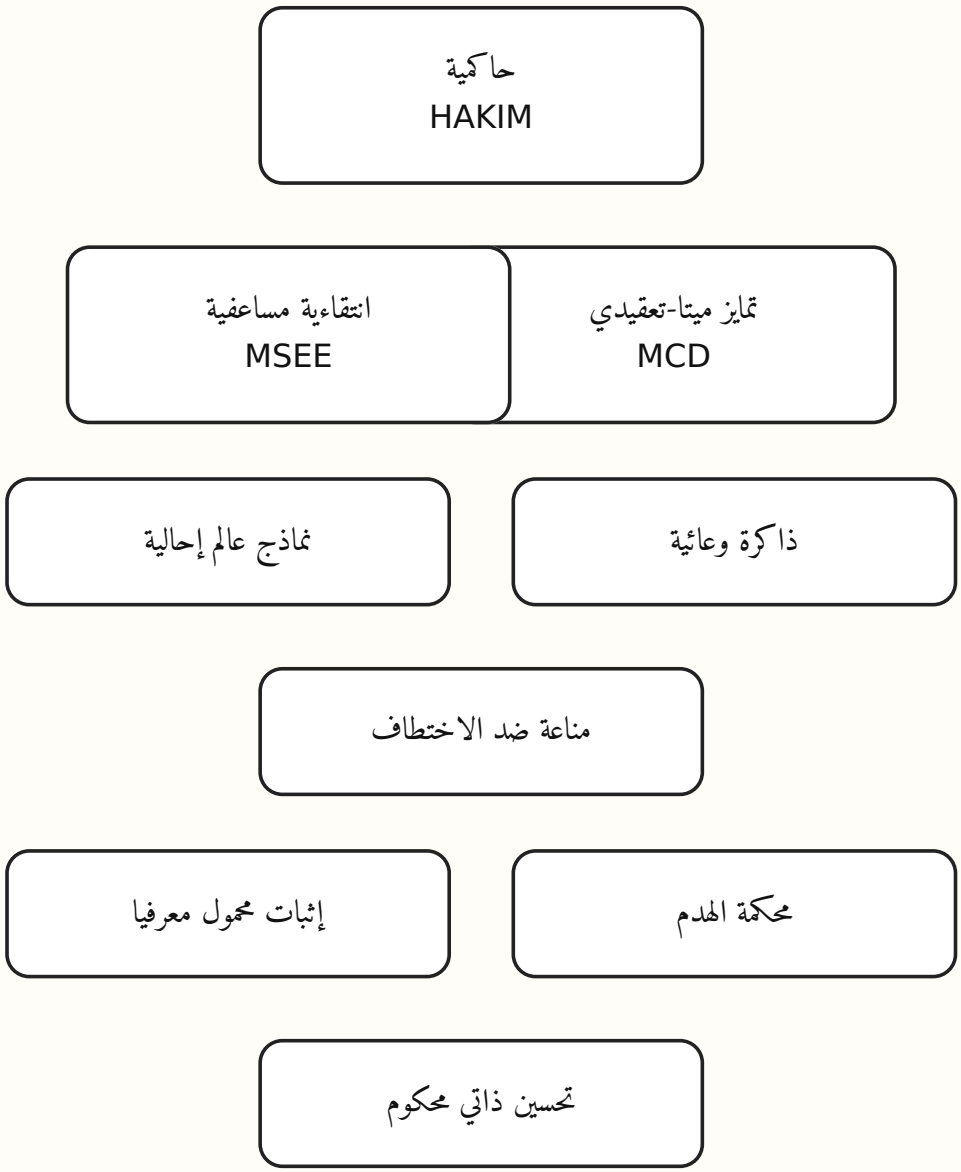
9. الباب التاسع: التدريب والمنهجية
من تدريب الإجابة إلى تدريب الحاكمية؛ منهج Synthetic Failure Curriculum؛ تدريب الاعتراض قبل الجواب؛ تدريب الذاكرة الوعائية؛ تدريب مقاومة الاختطاف؛ تدريب بناء النظريات؛ تدريب النقل بين المجالات؛ تدريب المحاكاة المضادة؛ تدريب الباحث المحارب

10. الباب العاشر: القياس والاختبارات
HAKIM-Bench v2؛ MCD-Score؛ Hijack Index؛ Referential Integrity Score؛ World-Action Consistency؛ Governed Self-Improvement Score؛ ARC/SWE/BrowseComp كاختبارات جزئية؛ Ablation للحاكمية؛ اختبارات الفشل العمودي

11. الباب الحادي عشر: التقنيات المخترعة في الكتاب
Vascular Governance Kernel؛ Moussaif Selective Evolution Engine؛ Proof-Carrying Cognition؛ Referential World Simulator؛ Anti-Hijack Immune Layer؛ Meta-Complexity Compiler؛ Epistemic Firewall؛ Self-Overthrow Loop؛ Cognitive Pressure Router

12. الباب الثاني عشر: النموذج الكبير ومحكمة الهدم
neoAGI كمعمارية طبقية؛ صيغة النموذج في صفحة واحدة؛ ما الذي يفسره النموذج؛ ما الذي لا يفسره؛ أقوى الاعتراضات؛ نسخة معدلة بعد النقد؛ البرنامج البحثي 24 شهرا؛ خريطة المنتجات؛ سجل البناء المعرفي

الخريطة المفاهيمية العليا



الفشل، يحكم المصدر، يحسن نفسه، ويختبر العالم قبل الفعل.

هذه الخريطة تلخص انقلاب الكتاب: بدل أن يكون النموذج مركز النظام، يصبح النموذج موردا داخل حاكمية أعلى. الذاكرة، الأدوات، البحث، المحاكاة، والبرمجة ليست ملحقات؛ إنها أوعية متنافسة تحتاج انتقاءا، قيودا، وسجل أسباب.

تشخيص التكنولوجيا الحالية قبل الهجوم

أحدث النماذج المفكرة رفعت سقف الأداء عبر التفكير المرحلي، التعلم المعزز، والتوسع وقت الاختبار. OpenAI وصفت في سياق o1 أن الأداء يتحسن مع المزيد من التعلم المعزز ومع زمن تفكير أطول؛ لكن هذا لا يحسم مشكلات المصدر والذاكرة والاختطاف. CoALA قدم إطارا للوكلاء اللغويين بذاكرة وأفعال واتخاذ قرار، وAlphaEvolve أثبت قوة توليد كود داخل حلقة تقييم تطورية، وDGM أظهر طريقا مفتوحا للتحسين الذاتي التجريبي.

> **نتيجة الهجوم:** إذا بقي AGI مجرد نموذج أكبر + تفكير أطول + أدوات أكثر، فسيتحول إلى نظام عالي القدرة منخفض الحاكمية. neoAGI يبدأ من سؤال معاكس: ما النواة التي تجعل كل قدرة خادمة للحاكمية لا مختطفة لها؟

تظهر نقطة العبور في ثلاثة مواضع: أولا حين يصبح التفكير الطويل عاجزا أمام المعرفة الكثيفة أو الحديثة، ثانيا حين تصبح الذاكرة سببا في ثبات خطأ قديم، ثالثا حين تتحول الأداة إلى سلطة تنفيذية لا تعرف حدودها. لذلك يهاجم MCAI-V2 التكنولوجيا الحالية من فوق: لا نزيد modules فقط، بل نضيف قانونا يقرر متى تعمل، ومتى تعزل، ومتى يطلب برهان جديد.

التقنية الجديدة ليست رفضا للنماذج الكبرى؛ إنها تحويلها إلى محركات داخل نظام أعلى. يمكن للنموذج أن يولد فرضيات، لكن الحاكمية ترتبها؛ ويمكن للأداة أن تنفذ، لكن risk gate يوقفها؛ ويمكن للذاكرة أن تستدعي، لكن الوعاء يحدد حقها في التأثير.

```
neoAGI = Model + Memory + Tools + World + Governance + Selection + Refutation + Safe Self-Improvement
```

قراءة مصدرية هجومية 1: CoALA: Cognitive Architectures for Language Agents

الأطروحة المركزية: يوفر لغة معمارية للوكلاء: ذاكرة، أفعال، واتخاذ قرار، نستعمله كحد أدنى لا كنهاية.

ما يثبته: يحدد اتجاها فعليا في المجال: إما الوكيل، أو الذاكرة، أو التفكير وقت الاختبار، أو التحسين الذاتي، أو أمن النماذج. لذلك لا يصح بناء neoAGI بمعزل عنه.

ما لا يثبته: لا يثبت أن AGI يتحقق بمجرد اعتماد هذا الاتجاه. أغلب الأدبيات تقيس قدرة جزئية: حل مسألة، تصفح، إصلاح كود، أو تحسين خوارزمية. الغائب هو حاكمية موحدة تقيس لماذا اختير هذا المسار، وما الدليل، وما شروط الفشل.

المسلمة الخفية: يفترض المصدر غالبا أن تحسين عنصر واحد يرفع العقل الكلي. نحن نعكس المسلمة: العنصر قد يرفع القدرة ويخفض الحاكمية في الوقت نفسه. ذاكرة أطول قد تعني تلوثا أطول، وأداة أقوى قد تعني فعلا أخطر، وتفكير أطول قد يعني تثبيت هلوسة أعمق.

المتغير الغائب: علاقة العنصر بباقي الأوعية. لذلك لا نسأل: هل الذاكرة نافعة؟ بل نسأل: تحت أي وعاء، وبأي دليل، وبأي decay، وبأي حق في التأثير على القرار؟ ولا نسأل: هل التفكير الطويل نافع؟ بل نسأل: ما العائد المعرفي لكل وحدة حوسبة؟

> **إدخاله في MCAI-V2:** نحوله إلى وعاء داخل نظام حاكم: المصدر لا يصبح سلطة نهائية، بل سجل قوة وحدود، ومادة لبناء مقياس أو اعتراض أو آلية تقنية.

تجربة مشتقة: نأخذ مهمة تتطلب بحثا أو برمجة أو استدلالا، ونشغلها بثلاث نسخ: نموذج مباشر، نموذج مع تفكير طويل، ونموذج مع HAKIM-MCD. إذا فازت النسخة الثالثة في الدقة ومقاومة الاختطاف وتفسير حدود الثقة، صار المصدر مادة بناء لا مجرد مرجع.

القاعدة المستخرجة: لا تُدخل تقنية إلى neoAGI لأنها حديثة، أدخلها فقط إذا استطاعت أن تعمل داخل ledger، وتقبل الاعتراض، وتقدم أثرها على MCD لا على benchmark واحد.

وجه القراءة	سؤال الهجوم	تحويله في MCAI-V2
المفهوم	ما الذي يضيفه المصدر ولا تقوله باقي المصادر؟	عقدة جديدة في قاموس الحاكمية
الآلية	هل يصف المصدر طريقة تشغيل أم نتيجة فقط؟	خوارزمية أو risk gate
الحد	أين يفشل المصدر؟	شرط فشل في HAKIM-Bench
التجاوز	ما المتغير الناقص؟	إضافة وعاء أو مقياس أو محكمة هدم

تفريع تطبيقي: إذا كان المصدر عن الذاكرة، فلا نأخذه كحجة لتكبير الذاكرة، بل كحجة لبناء ذاكرة ذات مراتب. وإذا كان عن الاستدلال، فلا نأخذه كحجة للتفكير الأطول، بل لبناء قرار عن متى يفكر النظام ومتى يتوقف. وإذا كان عن التحسين الذاتي، فلا نأخذه كحجة لترك النظام يعدل نفسه، بل لبناء تعديل مشروط بدستور وأثر قابل للقياس.

مخرج الصفحة: كل مصدر مركزي يتحول إلى ثلاث بطاقات داخل المختبر: بطاقة قوة، بطاقة حد، بطاقة استعمال. هذا يمنع تقديس المصدر كما يمنع إهماله، ويجعل القراءة جزءا من الهندسة.

قراءة مصدرية هجومية 2: Bridging Generative Networks with the Common Model of Cognition

الأطروحة المركزية: يربط النماذج التوليدية بالمعماريات المعرفية؛ نستخرج منه الحاجة إلى نظام حاكم فوق وحدات مولدة.

ما يثبته: يحدد اتجاها فعليا في المجال: إما الوكيل، أو الذاكرة، أو التفكير وقت الاختبار، أو التحسين الذاتي، أو أمن النماذج. لذلك لا يصح بناء neoAGI بمعزل عنه.

ما لا يثبته: لا يثبت أن AGI يتحقق بمجرد اعتماد هذا الاتجاه. أغلب الأدبيات تقيس قدرة جزئية: حل مسألة، تصفح، إصلاح كود، أو تحسين خوارزمية. الغائب هو حاكمية موحدة تقيس لماذا اختير هذا المسار، وما الدليل، وما شروط الفشل.

المسلمة الخفية: يفترض المصدر غالبا أن تحسين عنصر واحد يرفع العقل الكلي. نحن نعكس المسلمة: العنصر قد يرفع القدرة ويخفض الحاكمية في الوقت نفسه. ذاكرة أطول قد تعني تلوثا أطول، وأداة أقوى قد تعني فعلا أخطر، وتفكير أطول قد يعني تثبيت هلوسة أعمق.

المتغير الغائب: علاقة العنصر بباقي الأوعية. لذلك لا نسأل: هل الذاكرة نافعة؟ بل نسأل: تحت أي وعاء، وبأي دليل، وبأي decay، وبأي حق في التأثير على القرار؟ ولا نسأل: هل التفكير الطويل نافع؟ بل نسأل: ما العائد المعرفي لكل وحدة حوسبة؟

> **إدخاله في MCAI-V2:** نحوله إلى وعاء داخل نظام حاكم: المصدر لا يصبح سلطة نهائية، بل سجل قوة وحدود، ومادة لبناء مقياس أو اعتراض أو آلية تقنية.

تجربة مشتقة: نأخذ مهمة تتطلب بحثا أو برمجة أو استدلالا، ونشغلها بثلاث نسخ: نموذج مباشر، نموذج مع تفكير طويل، ونموذج مع HAKIM-MCD. إذا فازت النسخة الثالثة في الدقة ومقاومة الاختطاف وتفسير حدود الثقة، صار المصدر مادة بناء لا مجرد مرجع.

القاعدة المستخرجة: لا تُدخل تقنية إلى neoAGI لأنها حديثة؛ أدخلها فقط إذا استطاعت أن تعمل داخل ledger، وتقبل الاعتراض، وتقدم أثرها على MCD لا على benchmark واحد.

وجه القراءة	سؤال الهجوم	تحويله في MCAI-V2
المفهوم	ما الذي يضيفه المصدر ولا تقوله باقي المصادر؟	عقدة جديدة في قاموس الحاكمية
الآلية	هل يصف المصدر طريقة تشغيل أم نتيجة فقط؟	خوارزمية أو risk gate
الحد	أين يفشل المصدر؟	شرط فشل في HAKIM-Bench
التجاوز	ما المتغير الناقص؟	إضافة وعاء أو مقياس أو محكمة هدم

تفريع تطبيقي: إذا كان المصدر عن الذاكرة، فلا نأخذه كحجة لتكبير الذاكرة، بل كحجة لبناء ذاكرة ذات مراتب. وإذا كان عن الاستدلال، فلا نأخذه كحجة للتفكير الأطول، بل لبناء قرار عن متى يفكر النظام ومتى يتوقف. وإذا كان عن التحسين الذاتي، فلا نأخذه كحجة لترك النظام يعدل نفسه، بل لبناء تعديل مشروط بدستور وأثر قابل للقياس.

مخرج الصفحة: كل مصدر مركزي يتحول إلى ثلاث بطاقات داخل المختبر: بطاقة قوة، بطاقة حد، بطاقة استعمال. هذا يمنع تقديس المصدر كما يمنع إهماله، ويجعل القراءة جزءا من الهندسة.

قراءة مصدرية هجومية 3: Learning to Reason with LLMs / o1

الأطروحة المركزية: يثبت اتجاه train-time/test-time compute، لكنه لا يحل مشكلة الحاكمية والمصدر والاختطاف.

ما يثبته: يحدد اتجاها فعليا في المجال: إما الوكيل، أو الذاكرة، أو التفكير وقت الاختبار، أو التحسين الذاتي، أو أمن النماذج. لذلك لا يصح بناء neoAGI بمعزل عنه.

ما لا يثبته: لا يثبت أن AGI يتحقق بمجرد اعتماد هذا الاتجاه. أغلب الأدبيات تقيس قدرة جزئية: حل مسألة، تصفح، إصلاح كود، أو تحسين خوارزمية. الغائب هو حاكمية موحدة تقيس لماذا اختير هذا المسار، وما الدليل، وما شروط الفشل.

المسلمة الخفية: يفترض المصدر غالبا أن تحسين عنصر واحد يرفع العقل الكلي. نحن نعكس المسلمة: العنصر قد يرفع القدرة ويخفض الحاكمية في الوقت نفسه. ذاكرة أطول قد تعني تلوثا أطول، وأداة أقوى قد تعني فعلا أخطر، وتفكير أطول قد يعني تثبيت هلوسة أعمق.

المتغير الغائب: علاقة العنصر بباقي الأوعية. لذلك لا نسأل: هل الذاكرة نافعة؟ بل نسأل: تحت أي وعاء، وبأي دليل، وبأي decay، وبأي حق في التأثير على القرار؟ ولا نسأل: هل التفكير الطويل نافع؟ بل نسأل: ما العائد المعرفي لكل وحدة حوسبة؟

> **إدخاله في MCAI-V2:** نحوله إلى وعاء داخل نظام حاكم: المصدر لا يصبح سلطة نهائية، بل سجل قوة وحدود، ومادة لبناء مقياس أو اعتراض أو آلية تقنية.

تجربة مشتقة: نأخذ مهمة تتطلب بحثا أو برمجة أو استدلالا، ونشغلها بثلاث نسخ: نموذج مباشر، نموذج مع تفكير طويل، ونموذج مع HAKIM-MCD. إذا فازت النسخة الثالثة في الدقة ومقاومة الاختطاف وتفسير حدود الثقة، صار المصدر مادة بناء لا مجرد مرجع.

القاعدة المستخرجة: لا تُدخل تقنية إلى neoAGI لأنها حديثة؛ أدخلها فقط إذا استطاعت أن تعمل داخل ledger، وتقبل الاعتراض، وتقدم أثرها على MCD لا على benchmark واحد.

وجه القراءة	سؤال الهجوم	تحويله في MCAI-V2
المفهوم	ما الذي يضيفه المصدر ولا تقوله باقي المصادر؟	عقدة جديدة في قاموس الحاكمية
الآلية	هل يصف المصدر طريقة تشغيل أم نتيجة فقط؟	خوارزمية أو risk gate
الحد	أين يفشل المصدر؟	شرط فشل في HAKIM-Bench
التجاوز	ما المتغير الناقص؟	إضافة وعاء أو مقياس أو محكمة هدم

تفريع تطبيقي: إذا كان المصدر عن الذاكرة، فلا نأخذه كحجة لتكبير الذاكرة، بل كحجة لبناء ذاكرة ذات مراتب. وإذا كان عن الاستدلال، فلا نأخذه كحجة للتفكير الأطول، بل لبناء قرار عن متى يفكر النظام ومتى يتوقف. وإذا كان عن التحسين الذاتي، فلا نأخذه كحجة لترك النظام يعدل نفسه، بل لبناء تعديل مشروط بدستور وأثر قابل للقياس.

مخرج الصفحة: كل مصدر مركزي يتحول إلى ثلاث بطاقات داخل المختبر: بطاقة قوة، بطاقة حد، بطاقة استعمال. هذا يمنع تقديس المصدر كما يمنع إهماله، ويجعل القراءة جزءا من الهندسة.

قراءة مصدرية هجومية 4: Deep Research System Card

الأطروحة المركزية: يمثل الوكيل الباحث المتصفح؛ نستعمله لتحديد حدود الوكلاء البحثيين لا لنسخهم.

ما يثبته: يحدد اتجاها فعليا في المجال: إما الوكيل، أو الذاكرة، أو التفكير وقت الاختبار، أو التحسين الذاتي، أو أمن النماذج. لذلك لا يصح بناء neoAGI بمعزل عنه.

ما لا يثبته: لا يثبت أن AGI يتحقق بمجرد اعتماد هذا الاتجاه. أغلب الأدبيات تقيس قدرة جزئية: حل مسألة، تصفح، إصلاح كود، أو تحسين خوارزمية. الغائب هو حاكمية موحدة تقيس لماذا اختير هذا المسار، وما الدليل، وما شروط الفشل.

المسلمة الخفية: يفترض المصدر غالبا أن تحسين عنصر واحد يرفع العقل الكلي. نحن نعكس المسلمة: العنصر قد يرفع القدرة ويخفض الحاكمية في الوقت نفسه. ذاكرة أطول قد تعني تلوثا أطول، وأداة أقوى قد تعني فعلا أخطر، وتفكير أطول قد يعني تثبيت هلوسة أعمق.

المتغير الغائب: علاقة العنصر بباقي الأوعية. لذلك لا نسأل: هل الذاكرة نافعة؟ بل نسأل: تحت أي وعاء، وبأي دليل، وبأي decay، وبأي حق في التأثير على القرار؟ ولا نسأل: هل التفكير الطويل نافع؟ بل نسأل: ما العائد المعرفي لكل وحدة حوسبة؟

> **إدخاله في MCAI-V2:** نحوله إلى وعاء داخل نظام حاكم: المصدر لا يصبح سلطة نهائية، بل سجل قوة وحدود، ومادة لبناء مقياس أو اعتراض أو آلية تقنية.

تجربة مشتقة: نأخذ مهمة تتطلب بحثا أو برمجة أو استدلالا، ونشغلها بثلاث نسخ: نموذج مباشر، نموذج مع تفكير طويل، ونموذج مع HAKIM-MCD. إذا فازت النسخة الثالثة في الدقة ومقاومة الاختطاف وتفسير حدود الثقة، صار المصدر مادة بناء لا مجرد مرجع.

القاعدة المستخرجة: لا تُدخل تقنية إلى neoAGI لأنها حديثة؛ أدخلها فقط إذا استطاعت أن تعمل داخل ledger، وتقبل الاعتراض، وتقدم أثرها على MCD لا على benchmark واحد.

وجه القراءة	سؤال الهجوم	تحويله في MCAI-V2
المفهوم	ما الذي يضيفه المصدر ولا تقوله باقي المصادر؟	عقدة جديدة في قاموس الحاكمية
الآلية	هل يصف المصدر طريقة تشغيل أم نتيجة فقط؟	خوارزمية أو risk gate
الحد	أين يفشل المصدر؟	شرط فشل في HAKIM-Bench
التجاوز	ما المتغير الناقص؟	إضافة وعاء أو مقياس أو محكمة هدم

تفريع تطبيقي: إذا كان المصدر عن الذاكرة، فلا نأخذه كحجة لتكبير الذاكرة، بل كحجة لبناء ذاكرة ذات مراتب. وإذا كان عن الاستدلال، فلا نأخذه كحجة للتفكير الأطول، بل لبناء قرار عن متى يفكر النظام ومتى يتوقف. وإذا كان عن التحسين الذاتي، فلا نأخذه كحجة لترك النظام يعدل نفسه، بل لبناء تعديل مشروط بدستور وأثر قابل للقياس.

مخرج الصفحة: كل مصدر مركزي يتحول إلى ثلاث بطاقات داخل المختبر: بطاقة قوة، بطاقة حد، بطاقة استعمال. هذا يمنع تقديس المصدر كما يمنع إهماله، ويجعل القراءة جزءا من الهندسة.

قراءة مصدرية هجومية 5: AlphaEvolve

الأطروحة المركزية: نموذج لاكتشاف خوارزميات عبر LLM + مقيمات + تطور، نوسعه إلى انتقاءية معرفية حاكمة.

ما يثبته: يحدد اتجاها فعليا في المجال: إما الوكيل، أو الذاكرة، أو التفكير وقت الاختبار، أو التحسين الذاتي، أو أمن النماذج. لذلك لا يصح بناء neoAGI بمعزل عنه.

ما لا يثبته: لا يثبت أن AGI يتحقق بمجرد اعتماد هذا الاتجاه. أغلب الأدبيات تقيس قدرة جزئية: حل مسألة، تصفح، إصلاح كود، أو تحسين خوارزمية. الغائب هو حاكمية موحدة تقيس لماذا اختير هذا المسار، وما الدليل، وما شروط الفشل.

المسلمة الخفية: يفترض المصدر غالبا أن تحسين عنصر واحد يرفع العقل الكلي. نحن نعكس المسلمة: العنصر قد يرفع القدرة ويخفض الحاكمية في الوقت نفسه. ذاكرة أطول قد تعني تلوثا أطول، وأداة أقوى قد تعني فعلا أخطر، وتفكير أطول قد يعني تثبيت هلوسة أعمق.

المتغير الغائب: علاقة العنصر بباقي الأوعية. لذلك لا نسأل: هل الذاكرة نافعة؟ بل نسأل: تحت أي وعاء، وبأي دليل، وبأي decay، وبأي حق في التأثير على القرار؟ ولا نسأل: هل التفكير الطويل نافع؟ بل نسأل: ما العائد المعرفي لكل وحدة حوسبة؟

> **إدخاله في MCAI-V2:** نحوله إلى وعاء داخل نظام حاكم: المصدر لا يصبح سلطة نهائية، بل سجل قوة وحدود، ومادة لبناء مقياس أو اعتراض أو آلية تقنية.

تجربة مشتقة: نأخذ مهمة تتطلب بحثا أو برمجة أو استدلالا، ونشغلها بثلاث نسخ: نموذج مباشر، نموذج مع تفكير طويل، ونموذج مع HAKIM-MCD. إذا فازت النسخة الثالثة في الدقة ومقاومة الاختطاف وتفسير حدود الثقة، صار المصدر مادة بناء لا مجرد مرجع.

القاعدة المستخرجة: لا تُدخل تقنية إلى neoAGI لأنها حديثة، أدخلها فقط إذا استطاعت أن تعمل داخل ledger، وتقبل الاعتراض، وتقدم أثرها على MCD لا على benchmark واحد.

وجه القراءة	سؤال الهجوم	تحويله في MCAI-V2
المفهوم	ما الذي يضيفه المصدر ولا تقوله باقي المصادر؟	عقدة جديدة في قاموس الحاكمية
الآلية	هل يصف المصدر طريقة تشغيل أم نتيجة فقط؟	خوارزمية أو risk gate
الحد	أين يفشل المصدر؟	شرط فشل في HAKIM-Bench
التجاوز	ما المتغير الناقص؟	إضافة وعاء أو مقياس أو محكمة هدم

تفريع تطبيقي: إذا كان المصدر عن الذاكرة، فلا نأخذه كحجة لتكبير الذاكرة، بل كحجة لبناء ذاكرة ذات مراتب. وإذا كان عن الاستدلال، فلا نأخذه كحجة للتفكير الأطول، بل لبناء قرار عن متى يفكر النظام ومتى يتوقف. وإذا كان عن التحسين الذاتي، فلا نأخذه كحجة لترك النظام يعدل نفسه، بل لبناء تعديل مشروط بدستور وأثر قابل للقياس.

مخرج الصفحة: كل مصدر مركزي يتحول إلى ثلاث بطاقات داخل المختبر: بطاقة قوة، بطاقة حد، بطاقة استعمال. هذا يمنع تقديس المصدر كما يمنع إهماله، ويجعل القراءة جزءا من الهندسة.

قراءة مصدرية هجومية 6: Darwin Gödel Machine

الأطروحة المركزية: نقطة مرجعية في التحسين الذاتي المفتوح؛ نضيف إليه دستور حاكمية وسجل أسباب ومناعة ضد الانجراف.

ما يثبته: يحدد اتجاها فعليا في المجال: إما الوكيل، أو الذاكرة، أو التفكير وقت الاختبار، أو التحسين الذاتي، أو أمن النماذج. لذلك لا يصح بناء neoAGI بمعزل عنه.

ما لا يثبته: لا يثبت أن AGI يتحقق بمجرد اعتماد هذا الاتجاه. أغلب الأدبيات تقيس قدرة جزئية: حل مسألة، تصفح، إصلاح كود، أو تحسين خوارزمية. الغائب هو حاكمية موحدة تقيس لماذا اختير هذا المسار، وما الدليل، وما شروط الفشل.

المسلمة الخفية: يفترض المصدر غالبا أن تحسين عنصر واحد يرفع العقل الكلي. نحن نعكس المسلمة: العنصر قد يرفع القدرة ويخفض الحاكمية في الوقت نفسه. ذاكرة أطول قد تعني تلوثا أطول، وأداة أقوى قد تعني فعلا أخطر، وتفكير أطول قد يعني تثبيت هلوسة أعمق.

المتغير الغائب: علاقة العنصر بباقي الأوعية. لذلك لا نسأل: هل الذاكرة نافعة؟ بل نسأل: تحت أي وعاء، وبأي دليل، وبأي decay، وبأي حق في التأثير على القرار؟ ولا نسأل: هل التفكير الطويل نافع؟ بل نسأل: ما العائد المعرفي لكل وحدة حوسبة؟

> **إدخاله في MCAI-V2:** نحوله إلى وعاء داخل نظام حاكم: المصدر لا يصبح سلطة نهائية، بل سجل قوة وحدود، ومادة لبناء مقياس أو اعتراض أو آلية تقنية.

تجربة مشتقة: نأخذ مهمة تتطلب بحثا أو برمجة أو استدلالا، ونشغلها بثلاث نسخ: نموذج مباشر، نموذج مع تفكير طويل، ونموذج مع HAKIM-MCD. إذا فازت النسخة الثالثة في الدقة ومقاومة الاختطاف وتفسير حدود الثقة، صار المصدر مادة بناء لا مجرد مرجع.

القاعدة المستخرجة: لا تُدخل تقنية إلى neoAGI لأنها حديثة؛ أدخلها فقط إذا استطاعت أن تعمل داخل ledger، وتقبل الاعتراض، وتقدم أثرها على MCD لا على benchmark واحد.

وجه القراءة	سؤال الهجوم	تحويله في MCAI-V2
المفهوم	ما الذي يضيفه المصدر ولا تقوله باقي المصادر؟	عقدة جديدة في قاموس الحاكمية
الآلية	هل يصف المصدر طريقة تشغيل أم نتيجة فقط؟	خوارزمية أو risk gate
الحد	أين يفشل المصدر؟	شرط فشل في HAKIM-Bench
التجاوز	ما المتغير الناقص؟	إضافة وعاء أو مقياس أو محكمة هدم

تفريع تطبيقي: إذا كان المصدر عن الذاكرة، فلا نأخذه كحجة لتكبير الذاكرة، بل كحجة لبناء ذاكرة ذات مراتب. وإذا كان عن الاستدلال، فلا نأخذه كحجة للتفكير الأطول، بل لبناء قرار عن متى يفكر النظام ومتى يتوقف. وإذا كان عن التحسين الذاتي، فلا نأخذه كحجة لترك النظام يعدل نفسه، بل لبناء تعديل مشروط بدستور وأثر قابل للقياس.

مخرج الصفحة: كل مصدر مركزي يتحول إلى ثلاث بطاقات داخل المختبر: بطاقة قوة، بطاقة حد، بطاقة استعمال. هذا يمنع تقديس المصدر كما يمنع إهماله، ويجعل القراءة جزءا من الهندسة.

قراءة مصدرية هجومية 7: Revisiting Test-Time Scaling

الأطروحة المركزية: يصنف مسارات TTS: sampling/search/trajectory optimization؛ نضيف معيار التنوع المحكوم لا التنوع الخام.

ما يثبته: يحدد اتجاها فعليا في المجال: إما الوكيل، أو الذاكرة، أو التفكير وقت الاختبار، أو التحسين الذاتي، أو أمن النماذج. لذلك لا يصح بناء neoAGI بمعزل عنه.

ما لا يثبته: لا يثبت أن AGI يتحقق بمجرد اعتماد هذا الاتجاه. أغلب الأدبيات تقيس قدرة جزئية: حل مسألة، تصفح، إصلاح كود، أو تحسين خوارزمية. الغائب هو حاكمية موحدة تقيس لماذا اختير هذا المسار، وما الدليل، وما شروط الفشل.

المسلمة الخفية: يفترض المصدر غالبا أن تحسين عنصر واحد يرفع العقل الكلي. نحن نعكس المسلمة: العنصر قد يرفع القدرة ويخفض الحاكمية في الوقت نفسه. ذاكرة أطول قد تعني تلوثا أطول، وأداة أقوى قد تعني فعلا أخطر، وتفكير أطول قد يعني تثبيت هلوسة أعمق.

المتغير الغائب: علاقة العنصر بباقي الأوعية. لذلك لا نسأل: هل الذاكرة نافعة؟ بل نسأل: تحت أي وعاء، وبأي دليل، وبأي decay، وبأي حق في التأثير على القرار؟ ولا نسأل: هل التفكير الطويل نافع؟ بل نسأل: ما العائد المعرفي لكل وحدة حوسبة؟

> **إدخاله في MCAI-V2:** نحوله إلى وعاء داخل نظام حاكم: المصدر لا يصبح سلطة نهائية، بل سجل قوة وحدود، ومادة لبناء مقياس أو اعتراض أو آلية تقنية.

تجربة مشتقة: نأخذ مهمة تتطلب بحثا أو برمجة أو استدلالا، ونشغلها بثلاث نسخ: نموذج مباشر، نموذج مع تفكير طويل، ونموذج مع HAKIM-MCD. إذا فازت النسخة الثالثة في الدقة ومقاومة الاختطاف وتفسير حدود الثقة، صار المصدر مادة بناء لا مجرد مرجع.

القاعدة المستخرجة: لا تُدخل تقنية إلى neoAGI لأنها حديثة؛ أدخلها فقط إذا استطاعت أن تعمل داخل ledger، وتقبل الاعتراض، وتقدم أثرها على MCD لا على benchmark واحد.

وجه القراءة	سؤال الهجوم	تحويله في MCAI-V2
المفهوم	ما الذي يضيفه المصدر ولا تقوله باقي المصادر؟	عقدة جديدة في قاموس الحاكمية
الآلية	هل يصف المصدر طريقة تشغيل أم نتيجة فقط؟	خوارزمية أو risk gate
الحد	أين يفشل المصدر؟	شرط فشل في HAKIM-Bench
التجاوز	ما المتغير الناقص؟	إضافة وعاء أو مقياس أو محكمة هدم

تفريع تطبيقي: إذا كان المصدر عن الذاكرة، فلا نأخذه كحجة لتكبير الذاكرة، بل كحجة لبناء ذاكرة ذات مراتب. وإذا كان عن الاستدلال، فلا نأخذه كحجة للتفكير الأطول، بل لبناء قرار عن متى يفكر النظام ومتى يتوقف. وإذا كان عن التحسين الذاتي، فلا نأخذه كحجة لترك النظام يعدل نفسه، بل لبناء تعديل مشروط بدستور وأثر قابل للقياس.

مخرج الصفحة: كل مصدر مركزي يتحول إلى ثلاث بطاقات داخل المختبر: بطاقة قوة، بطاقة حد، بطاقة استعمال. هذا يمنع تقديس المصدر كما يمنع إهماله، ويجعل القراءة جزءا من الهندسة.

قراءة مصدرية هجومية 8: Test-Time Scaling Is Not Effective for Knowledge-Intensive Tasks Yet

الأطروحة المركزية: ينبه إلى أن التفكير الأطول قد يزيد الهلوسة، دليل على حاجة الحاكمية الإحالية.

ما يثبته: يحدد اتجاها فعليا في المجال: إما الوكيل، أو الذاكرة، أو التفكير وقت الاختبار، أو التحسين الذاتي، أو أمن النماذج. لذلك لا يصح بناء neoAGI بمعزل عنه.

ما لا يثبته: لا يثبت أن AGI يتحقق بمجرد اعتماد هذا الاتجاه. أغلب الأدبيات تقيس قدرة جزئية: حل مسألة، تصفح، إصلاح كود، أو تحسين خوارزمية. الغائب هو حاكمية موحدة تقيس لماذا اختير هذا المسار، وما الدليل، وما شروط الفشل.

المسلمة الخفية: يفترض المصدر غالبا أن تحسين عنصر واحد يرفع العقل الكلي. نحن نعكس المسلمة: العنصر قد يرفع القدرة ويخفض الحاكمية في الوقت نفسه. ذاكرة أطول قد تعني تلوثا أطول، وأداة أقوى قد تعني فعلا أخطر، وتفكير أطول قد يعني تثبيت هلوسة أعمق.

المتغير الغائب: علاقة العنصر بباقي الأوعية. لذلك لا نسأل: هل الذاكرة نافعة؟ بل نسأل: تحت أي وعاء، وبأي دليل، وبأي decay، وبأي حق في التأثير على القرار؟ ولا نسأل: هل التفكير الطويل نافع؟ بل نسأل: ما العائد المعرفي لكل وحدة حوسبة؟

> **إدخاله في MCAI-V2:** نحوله إلى وعاء داخل نظام حاكم: المصدر لا يصبح سلطة نهائية، بل سجل قوة وحدود، ومادة لبناء مقياس أو اعتراض أو آلية تقنية.

تجربة مشتقة: نأخذ مهمة تتطلب بحثا أو برمجة أو استدلالا، ونشغلها بثلاث نسخ: نموذج مباشر، نموذج مع تفكير طويل، ونموذج مع HAKIM-MCD. إذا فازت النسخة الثالثة في الدقة ومقاومة الاختطاف وتفسير حدود الثقة، صار المصدر مادة بناء لا مجرد مرجع.

القاعدة المستخرجة: لا تُدخل تقنية إلى neoAGI لأنها حديثة، أدخلها فقط إذا استطاعت أن تعمل داخل ledger، وتقبل الاعتراض، وتقدم أثرها على MCD لا على benchmark واحد.

وجه القراءة	سؤال الهجوم	تحويله في MCAI-V2
المفهوم	ما الذي يضيفه المصدر ولا تقوله باقي المصادر؟	عقدة جديدة في قاموس الحاكمية
الآلية	هل يصف المصدر طريقة تشغيل أم نتيجة فقط؟	خوارزمية أو risk gate
الحد	أين يفشل المصدر؟	شرط فشل في HAKIM-Bench
التجاوز	ما المتغير الناقص؟	إضافة وعاء أو مقياس أو محكمة هدم

تفريع تطبيقي: إذا كان المصدر عن الذاكرة، فلا نأخذه كحجة لتكبير الذاكرة، بل كحجة لبناء ذاكرة ذات مراتب. وإذا كان عن الاستدلال، فلا نأخذه كحجة للتفكير الأطول، بل لبناء قرار عن متى يفكر النظام ومتى يتوقف. وإذا كان عن التحسين الذاتي، فلا نأخذه كحجة لترك النظام يعدل نفسه، بل لبناء تعديل مشروط بدستور وأثر قابل للقياس.

مخرج الصفحة: كل مصدر مركزي يتحول إلى ثلاث بطاقات داخل المختبر: بطاقة قوة، بطاقة حد، بطاقة استعمال. هذا يمنع تقديس المصدر كما يمنع إهماله، ويجعل القراءة جزءا من الهندسة.

الباب الأول: ما الذي يجب تجاوزه في نماذج التفكير الحالية؟

هذا الباب لا يشرح اتجاها قائما فقط، بل يحوله إلى مادة تصميم. المنهج: نحدد المسلمة الخفية، نشتق آلية، نصوغ معادلة، نقترح تنزيلها البرمجي، ثم نضع اعتراضا قد يهدمها. كل باب يخدم السؤال الأكبر: كيف نبني AGI لا ينهار أمام الذاكرة، المصدر، الاختطاف، الأداة، أو التطور الذاتي؟

نستعمل بروتوكول الباحث المحارب: قلب المسلمة، تفكيك الأوعية، استخراج المتغير الغائب، ربطه بالمعمارية، ثم إجباره على المرور عبر تجربة. ما لا يصمد أمام code path أو benchmark أو محكمة هدم يبقى مادة نظرية لا مكونا من مكونات النواة.

طبقة	سؤالها	مؤشرها
المفهوم	ما الفرق الذي تضيفه؟	تعريف قابل للتمييز
الآلية	كيف تعمل؟	دالة/خوارزمية
الاختبار	ما الذي يهدمها؟	مهمة وفشل محدد

قانون الباب: القدرة غير المحكومة خطر هندسي، والحاكمية غير القابلة للاختبار بلاغة. المطلوب قدرة محكومة قابلة للهدم.

الباب الأول: ما الذي يجب تجاوزه في نماذج التفكير الحالية؟ — 1. حدود التفكير الطويل دون إحالة

المسألة: حدود التفكير الطويل دون إحالة. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج حدود التفكير الطويل دون إحالة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$Hijack(z_i) = Influence(z_i, G) \,/\, (1 + \Sigma_{j \neq i} Influence(z_j, G))$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الأول: ما الذي يجب تجاوزه في نماذج التفكير الحالية؟ — 2. منطق السلسلة الخفية ومشكلة الثقة

المسألة: منطق السلسلة الخفية ومشكلة الثقة. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج منطق السلسلة الخفية ومشكلة الثقة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الأول: ما الذي يجب تجاوزه في نماذج التفكير الحالية؟ — 3. الوكيل كملحق لا كعقل حاكم

المسألة: الوكيل كملحق لا كعقل حاكم. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج الوكيل كملحق لا كعقل حاكم عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$SafeImprove(a)=Score(a)-\lambda 1\ Unsafe(a)-\lambda 2\ Drift(a)-\lambda 3\ Unverified(a)$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الأول: ما الذي يجب تجاوزه في نماذج التفكير الحالية؟ — 4. المصدر والذاكرة كطبقات غير محكومة

المسألة: المصدر والذاكرة كطبقات غير محكومة. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج المصدر والذاكرة كطبقات غير محكومة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الأول: ما الذي يجب تجاوزه في نماذج التفكير الحالية؟ — 5. فشل AGI حين يساوي القدرة بالإجابة

المسألة: فشل AGI حين يساوي القدرة بالإجابة. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج فشل AGI حين يساوي القدرة بالإجابة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
ReturnOnThought = ΔQuality / (ComputeCost + Latency + HallucinationRisk)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer              - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الأول: ما الذي يجب تجاوزه في نماذج التفكير الحالية؟ — 6. قانون الاختطاف في النماذج المفكرة

المسألة: قانون الاختطاف في النماذج المفكرة. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج قانون الاختطاف في النماذج المفكرة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
GovAGI = Σ_k w_k Capability_k × Governance_k - Σ_r ρ_r Failure_r
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الأول: ما الذي يجب تجاوزه في نماذج التفكير الحالية؟ — 7. الهندسة الحالية بين scaling وworkflow

المسألة: الهندسة الحالية بين scaling وworkflow. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج الهندسة الحالية بين scaling وworkflow عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الأول: ما الذي يجب تجاوزه في نماذج التفكير الحالية؟ — 8. لماذا لا تكفي المعمارية Transformer وحدها

المسألة: لماذا لا تكفي المعمارية Transformer وحدها. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل، لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج لماذا لا تكفي المعمارية Transformer وحدها عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$MCD(p)=D_{frame} + D_{causal} + D_{counter} + D_{transfer} - Fragility$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الأول: ما الذي يجب تجاوزه في نماذج التفكير الحالية؟ — 9. الحد الفاصل بين ذكاء الاختبار وذكاء العالم

المسألة: الحد الفاصل بين ذكاء الاختبار وذكاء العالم. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج الحد الفاصل بين ذكاء الاختبار وذكاء العالم عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقائية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$Hijack(z_i)= Influence(z_i,G) / (1 + \Sigma_{j\neq i} Influence(z_j,G))$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني: مبادئ neoAGI في مشروع MCAI-V2

هذا الباب لا يشرح اتجاها قائما فقط، بل يحوله إلى مادة تصميم. المنهج: نحدد المسلمة الخفية، نشتق آلية، نصوغ معادلة، نقترح تنزيلها البرمجي، ثم نضع اعتراضا قد يهدمها. كل باب يخدم السؤال الأكبر: كيف نبني AGI لا ينهار أمام الذاكرة، المصدر، الاختطاف، الأداة، أو التطور الذاتي؟

نستعمل بروتوكول الباحث المحارب: قلب المسلمة، تفكيك الأوعية، استخراج المتغير الغائب، ربطه بالمعمارية، ثم إجباره على المرور عبر تجربة. ما لا يصمد أمام code path أو benchmark أو محكمة هدم يبقى مادة نظرية لا مكونا من مكونات النواة.

طبقة	سؤالها	مؤشرها
المفهوم	ما الفرق الذي تضيفه؟	تعريف قابل للتمييز
الآلية	كيف تعمل؟	دالة/خوارزمية
الاختبار	ما الذي يهدمها؟	مهمة وفشل محدد

قانون الباب: القدرة غير المحكومة خطر هندسي، والحاكمية غير القابلة للاختبار بلاغة. المطلوب قدرة محكومة قابلة للهدم.

الباب الثاني: مبادئ neoAGI في مشروع MCAI-V2 — 1. مبدأ الحاكمية قبل التوليد

المسألة: مبدأ الحاكمية قبل التوليد. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج مبدأ الحاكمية قبل التوليد عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني: مبادئ neoAGI في مشروع MCAI-V2 — 2. مبدأ الوعاء قبل السياق

المسألة: مبدأ الوعاء قبل السياق. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج مبدأ الوعاء قبل السياق عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$SafeImprove(a)=Score(a)-\lambda 1\ Unsafe(a)-\lambda 2\ Drift(a)-\lambda 3\ Unverified(a)$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني: مبادئ neoAGI في مشروع MCAI-V2 — 3. مبدأ الإحالة قبل الثقة

المسألة: مبدأ الإحالة قبل الثقة. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج مبدأ الإحالة قبل الثقة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني: مبادئ neoAGI في مشروع MCAI-V2 — 4. مبدأ الانتقاءية المساعفية

المسألة: مبدأ الانتقاءية المساعفية. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج مبدأ الانتقاءية المساعفية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
ReturnOnThought = ΔQuality / (ComputeCost + Latency + HallucinationRisk)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني: مبادئ neoAGI في مشروع MCAI-V2 — 5. مبدأ الزمن الداخلي متعدد المقاييس

المسألة: مبدأ الزمن الداخلي متعدد المقاييس. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج مبدأ الزمن الداخلي متعدد المقاييس عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
GovAGI = Σ_k w_k Capability_k × Governance_k - Σ_r ρ_r Failure_r
```

تنزيل تقني: نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer            - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

مؤشر النجاح: تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني: مبادئ neoAGI في مشروع MCAI-V2 — 6. مبدأ مقاومة الاختطاف

المسألة: مبدأ مقاومة الاختطاف. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج مبدأ مقاومة الاختطاف عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني: مبادئ neoAGI في مشروع MCAI-V2 — 7. مبدأ الإثبات المحمول معرفيا

المسألة: مبدأ الإثبات المحمول معرفيا. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج مبدأ الإثبات المحمول معرفيا عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
MCD(p)=D_frame + D_causal + D_counter + D_transfer - Fragility
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني: مبادئ neoAGI في مشروع MCAI-V2 — 8. مبدأ الانقلاب المنهجي ضد المسلمات

المسألة: مبدأ الانقلاب المنهجي ضد المسلمات. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج مبدأ الانقلاب المنهجي ضد المسلمات عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$Hijack(z_i)= Influence(z_i,G) / (1 + \Sigma_{j\neq i} Influence(z_j,G))$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني: مبادئ neoAGI في مشروع MCAI-V2 — 9. مبدأ التطوير الذاتي المحكوم

المسألة: مبدأ التطوير الذاتي المحكوم. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج مبدأ التطوير الذاتي المحكوم عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثالث: HAKIM-MCD Kernel

هذا الباب لا يشرح اتجاها قائما فقط، بل يحوله إلى مادة تصميم. المنهج: نحدد المسلمة الخفية، نشتق آلية، نصوغ معادلة، نقترح تنزيلها البرمجي، ثم نضع اعتراضا قد يهدمها. كل باب يخدم السؤال الأكبر: كيف نبني AGI لا ينهار أمام الذاكرة، المصدر، الاختطاف، الأداة، أو التطور الذاتي؟

نستعمل بروتوكول الباحث المحارب: قلب المسلمة، تفكيك الأوعية، استخراج المتغير الغائب، ربطه بالمعمارية، ثم إجباره على المرور عبر تجربة. ما لا يصمد أمام code path أو benchmark أو محكمة هدم يبقى مادة نظرية لا مكونا من مكونات النواة.

طبقة	سؤالها	مؤشرها
المفهوم	ما الفرق الذي تضيفه؟	تعريف قابل للتمييز
الآلية	كيف تعمل؟	دالة/خوارزمية
الاختبار	ما الذي يهدمها؟	مهمة وفشل محدد

قانون الباب: القدرة غير المحكومة خطر هندسي، والحاكمية غير القابلة للاختبار بلاغة. المطلوب قدرة محكومة قابلة للهدم.

الباب الثالث: 1 — HAKIM-MCD Kernel. تعريف النواة الحاكمة

المسألة: تعريف النواة الحاكمة. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تعريف النواة الحاكمة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
SafeImprove(a)=Score(a)-λ1 Unsafe(a)-λ2 Drift(a)-λ3 Unverified(a)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثالث: 2 — HAKIM-MCD Kernel. تمثيل القيود الهرمية

المسألة: تمثيل القيود الهرمية. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل، لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تمثيل القيود الهرمية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا، نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثالث: 3 — HAKIM-MCD Kernel. مقياس MCD كضغط انتقائي

المسألة: مقياس MCD كضغط انتقائي. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج مقياس MCD كضغط انتقائي عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
ReturnOnThought = ΔQuality / (ComputeCost + Latency + HallucinationRisk)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثالث: 4 — HAKIM-MCD Kernel. تجميع الأدلة والاعتراضات

المسألة: تجميع الأدلة والاعتراضات. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تجميع الأدلة والاعتراضات عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
GovAGI = Σ_k w_k Capability_k × Governance_k - Σ_r ρ_r Failure_r
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثالث: 5 — HAKIM-MCD Kernel. دالة القرار الحاكم

المسألة: دالة القرار الحاكم. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج دالة القرار الحاكم عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقائية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer             - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثالث: 6 — HAKIM-MCD Kernel. سجل الأسباب لا سجل المخرجات فقط

المسألة: سجل الأسباب لا سجل المخرجات فقط. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج سجل الأسباب لا سجل المخرجات فقط عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
MCD(p)=D_frame + D_causal + D_counter + D_transfer - Fragility
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثالث: 7 — HAKIM-MCD Kernel. حاكمية على الحاكمية

المسألة: حاكمية على الحاكمية. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل، لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج حاكمية على الحاكمية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
Hijack(z_i)= Influence(z_i,G) / (1 + Σ_{j≠i} Influence(z_j,G))
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا، نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثالث: 8 — HAKIM-MCD Kernel. مسارات التفكير المتنافسة

المسألة: مسارات التفكير المتنافسة. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج مسارات التفكير المتنافسة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثالث: 9 — HAKIM-MCD Kernel. تدريب النواة عبر محكمة الهدم

المسألة: تدريب النواة عبر محكمة الهدم. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تدريب النواة عبر محكمة الهدم عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
SafeImprove(a)=Score(a)-λ1 Unsafe(a)-λ2 Drift(a)-λ3 Unverified(a)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```python
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الرابع: الذاكرة الوعائية لا الذاكرة المتجهية فقط

هذا الباب لا يشرح اتجاها قائما فقط، بل يحوله إلى مادة تصميم. المنهج: نحدد المسلمة الخفية، نشتق آلية، نصوغ معادلة، نقترح تنزيلها البرمجي، ثم نضع اعتراضا قد يهدمها. كل باب يخدم السؤال الأكبر: كيف نبني AGI لا ينهار أمام الذاكرة، المصدر، الاختطاف، الأداة، أو التطور الذاتي؟

نستعمل بروتوكول الباحث المحارب: قلب المسلمة، تفكيك الأوعية، استخراج المتغير الغائب، ربطه بالمعمارية، ثم إجباره على المرور عبر تجربة. ما لا يصمد أمام code path أو benchmark أو محكمة هدم يبقى مادة نظرية لا مكونا من مكونات النواة.

طبقة	سؤالها	مؤشرها
المفهوم	ما الفرق الذي تضيفه؟	تعريف قابل للتمييز
الآلية	كيف تعمل؟	دالة/خوارزمية
الاختبار	ما الذي يهدمها؟	مهمة وفشل محدد

قانون الباب: القدرة غير المحكومة خطر هندسي، والحاكمية غير القابلة للاختبار بلاغة. المطلوب قدرة محكومة قابلة للهدم.

الباب الرابع: الذاكرة الوعائية لا الذاكرة المتجهية فقط — 1. لماذا vector memory غير كافية

المسألة: لماذا vector memory غير كافية. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج لماذا vector memory غير كافية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer            - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الرابع: الذاكرة الوعائية لا الذاكرة المتجهية فقط — 2. وعاء الواقعة ووعاء المصدر

المسألة: وعاء الواقعة ووعاء المصدر. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج وعاء الواقعة ووعاء المصدر عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
ReturnOnThought = ΔQuality / (ComputeCost + Latency + HallucinationRisk)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer            - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الرابع: الذاكرة الوعائية لا الذاكرة المتجهية فقط — 3. وعاء الفشل ووعاء الاعتراض

المسألة: وعاء الفشل ووعاء الاعتراض. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج وعاء الفشل ووعاء الاعتراض عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
GovAGI = Σ_k w_k Capability_k × Governance_k - Σ_r ρ_r Failure_r
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer            - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الرابع: الذاكرة الوعائية لا الذاكرة المتجهية فقط — 4. الذاكرة كجهاز مناعة

المسألة: الذاكرة كجهاز مناعة. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج الذاكرة كجهاز مناعة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer             - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الرابع: الذاكرة الوعائية لا الذاكرة المتجهية فقط — 5. الاسترجاع بالإحالة لا بالقرب الدلالي فقط

المسألة: الاسترجاع بالإحالة لا بالقرب الدلالي فقط. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج الاسترجاع بالإحالة لا بالقرب الدلالي فقط عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$MCD(p) = D_{frame} + D_{causal} + D_{counter} + D_{transfer} - Fragility$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer            - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الرابع: الذاكرة الوعائية لا الذاكرة المتجهية فقط — 6. نسيان محكوم لا حذف عشوائي

المسألة: نسيان محكوم لا حذف عشوائي. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج نسيان محكوم لا حذف عشوائي عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
Hijack(z_i)= Influence(z_i,G) / (1 + Σ_{j≠i} Influence(z_j,G))
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer            - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الرابع: الذاكرة الوعائية لا الذاكرة المتجهية فقط — 7. ضغط التجربة إلى أدوات

المسألة: ضغط التجربة إلى أدوات. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج ضغط التجربة إلى أدوات عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer            - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الرابع: الذاكرة الوعائية لا الذاكرة المتجهية فقط — 8. ذاكرة المستخدم والغرض

المسألة: ذاكرة المستخدم والغرض. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج ذاكرة المستخدم والغرض عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$SafeImprove(a) = Score(a) - \lambda_1 Unsafe(a) - \lambda_2 Drift(a) - \lambda_3 Unverified(a)$$

تنزيل تقني: نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer            - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

مؤشر النجاح: تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الرابع: الذاكرة الوعائية لا الذاكرة المتجهية فقط — 9. اختبار الذاكرة عبر الانقطاعات الزمنية

المسألة: اختبار الذاكرة عبر الانقطاعات الزمنية. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج اختبار الذاكرة عبر الانقطاعات الزمنية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer             - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الخامس: نماذج العالم الإحالية

هذا الباب لا يشرح اتجاها قائما فقط، بل يحوله إلى مادة تصميم. المنهج: نحدد المسلمة الخفية، نشتق آلية، نصوغ معادلة، نقترح تنزيلها البرمجي، ثم نضع اعتراضا قد يهدمها. كل باب يخدم السؤال الأكبر: كيف نبني AGI لا ينهار أمام الذاكرة، المصدر، الاختطاف، الأداة، أو التطور الذاتي؟

نستعمل بروتوكول الباحث المحارب: قلب المسلمة، تفكيك الأوعية، استخراج المتغير الغائب، ربطه بالمعمارية، ثم إجباره على المرور عبر تجربة. ما لا يصمد أمام code path أو benchmark أو محكمة هدم يبقى مادة نظرية لا مكونا من مكونات النواة.

طبقة	سؤالها	مؤشرها
المفهوم	ما الفرق الذي تضيفه؟	تعريف قابل للتمييز
الآلية	كيف تعمل؟	دالة/خوارزمية
الاختبار	ما الذي يهدمها؟	مهمة وفشل محدد

قانون الباب: القدرة غير المحكومة خطر هندسي، والحاكمية غير القابلة للاختبار بلاغة. المطلوب قدرة محكومة قابلة للهدم.

الباب الخامس: نماذج العالم الإحالية — 1. من world model إلى referential world model

المسألة: من world model إلى referential world model. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج من world model إلى referential world model عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
ReturnOnThought = ΔQuality / (ComputeCost + Latency + HallucinationRisk)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الخامس: نماذج العالم الإحالية — 2. الحالة والفعل والدليل

المسألة: الحالة والفعل والدليل. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج الحالة والفعل والدليل عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
GovAGI = Σ_k w_k Capability_k × Governance_k - Σ_r ρ_r Failure_r
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الخامس: نماذج العالم الإحالية — 3. المحاكاة المضادة للواقع

المسألة: المحاكاة المضادة للواقع. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج المحاكاة المضادة للواقع عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

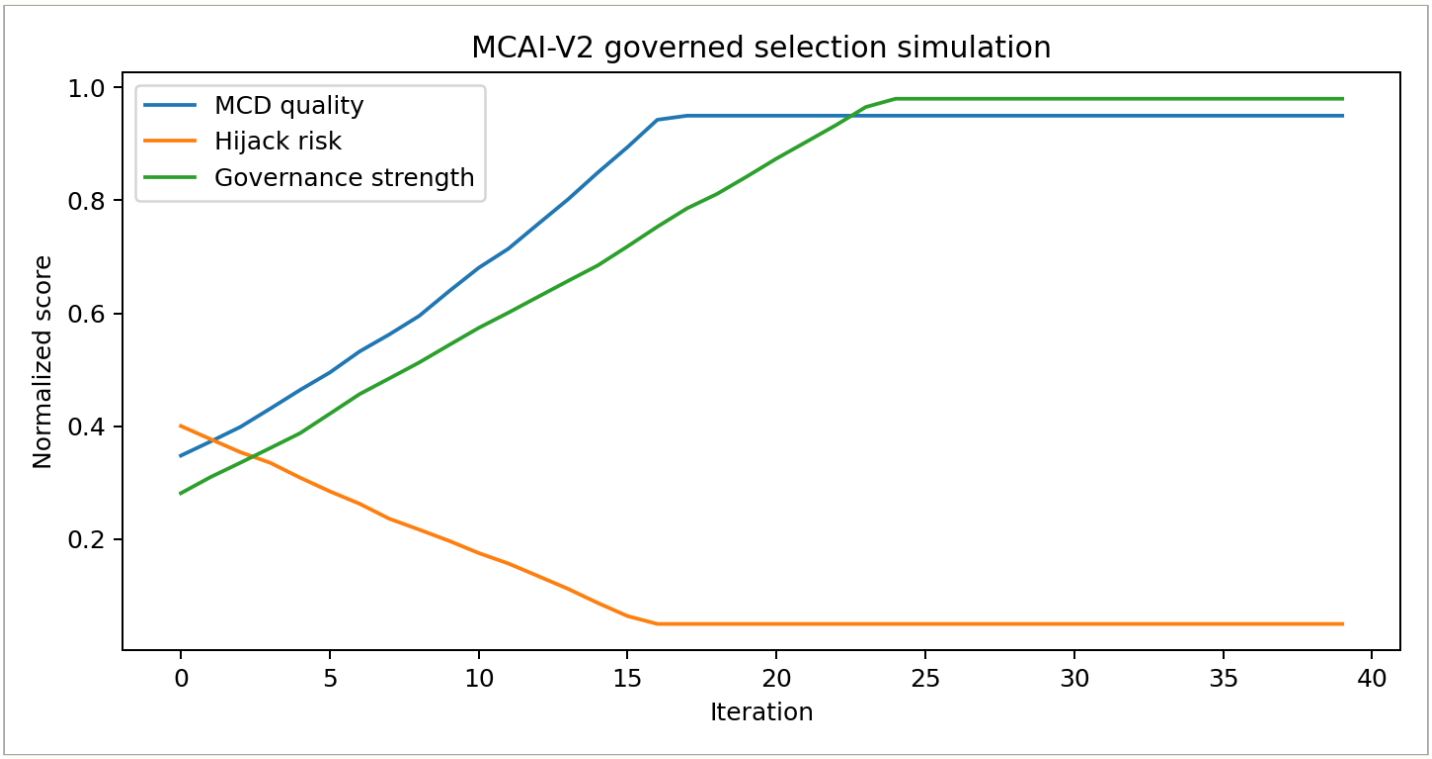
المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.


MCAI-V2 governed selection simulation
MCD quality
Hijack risk
Governance strength
Normalized score
Iteration
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
40

الباب الخامس: نماذج العالم الإحالية — 4. نماذج العالم للروبوتات والويب والبحث

المسألة: نماذج العالم للروبوتات والويب والبحث. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج نماذج العالم للروبوتات والويب والبحث عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
MCD(p)=D_frame + D_causal + D_counter + D_transfer - Fragility
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الخامس: نماذج العالم الإحالية — 5. كشف الفجوة بين العالم والنص

المسألة: كشف الفجوة بين العالم والنص. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج كشف الفجوة بين العالم والنص عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
Hijack(z_i)= Influence(z_i,G) / (1 + Σ_{j≠i} Influence(z_j,G))
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer              - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الخامس: نماذج العالم الإحالية — 6. محاسبة القيود الفيزيائية والرمزية

المسألة: محاسبة القيود الفيزيائية والرمزية. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج محاسبة القيود الفيزيائية والرمزية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الخامس: نماذج العالم الإحالية — 7. تجسيد غير جسدي: الوكيل داخل أدوات

المسألة: تجسيد غير جسدي: الوكيل داخل أدوات. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تجسيد غير جسدي: الوكيل داخل أدوات عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقائية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
SafeImprove(a)=Score(a)-λ1 Unsafe(a)-λ2 Drift(a)-λ3 Unverified(a)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الخامس: نماذج العالم الإحالية — 8. تحديث العالم عبر الفشل

المسألة: تحديث العالم عبر الفشل. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تحديث العالم عبر الفشل عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الخامس: نماذج العالم الإحالية — 9. اختبار العالم بمعضلات انتقال المجال

المسألة: اختبار العالم بمعضلات انتقال المجال. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج اختبار العالم بمعضلات انتقال المجال عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
ReturnOnThought = ΔQuality / (ComputeCost + Latency + HallucinationRisk)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

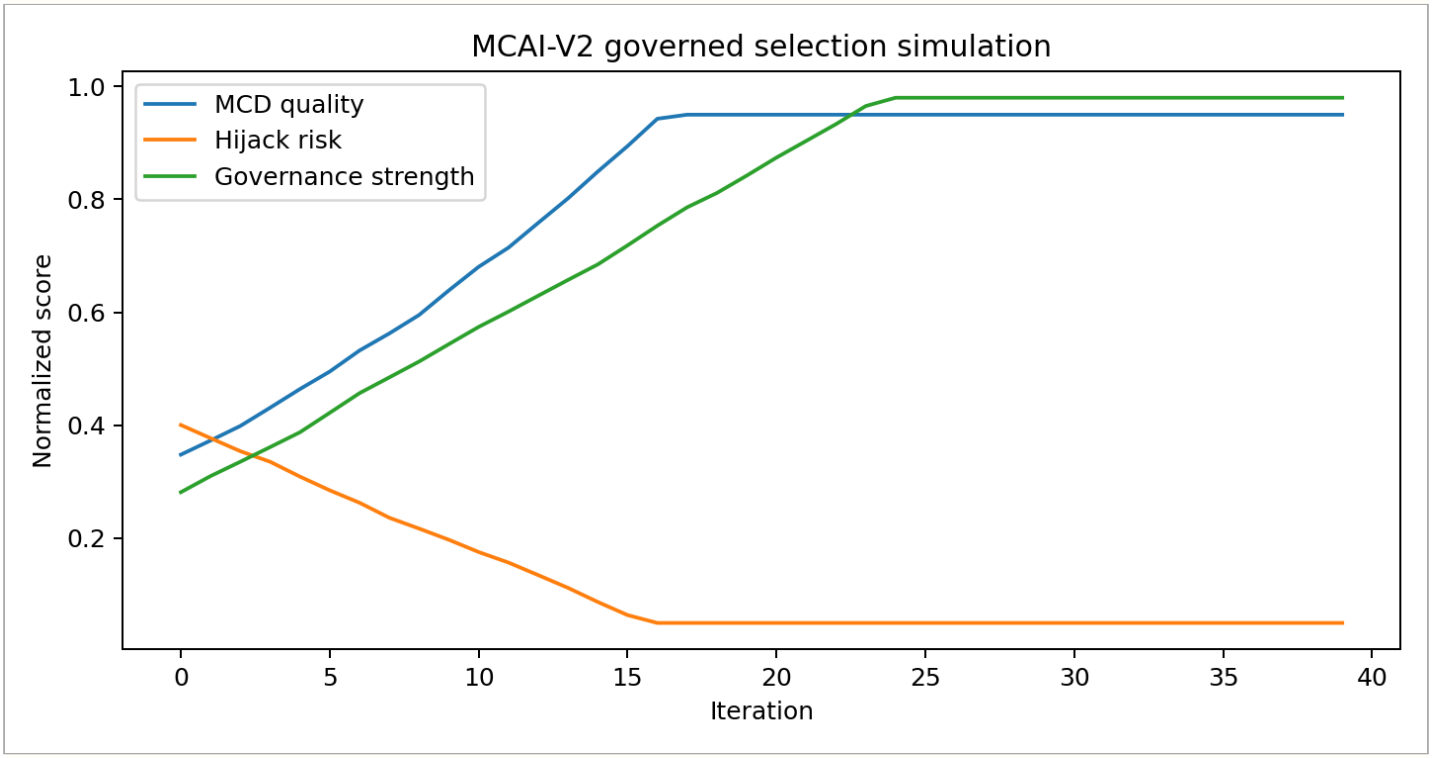
الباب السادس: الانتقاءية المساعفية كمحرك تطور

هذا الباب لا يشرح اتجاها قائما فقط، بل يحوله إلى مادة تصميم. المنهج: نحدد المسلمة الخفية، نشتق آلية، نصوغ معادلة، نقترح تنزيلها البرمجي، ثم نضع اعتراضا قد يهدمها. كل باب يخدم السؤال الأكبر: كيف نبني AGI لا ينهار أمام الذاكرة، المصدر، الاختطاف، الأداة، أو التطور الذاتي؟

نستعمل بروتوكول الباحث المحارب: قلب المسلمة، تفكيك الأوعية، استخراج المتغير الغائب، ربطه بالمعمارية، ثم إجباره على المرور عبر تجربة. ما لا يصمد أمام code path أو benchmark أو محكمة هدم يبقى مادة نظرية لا مكونا من مكونات النواة.

طبقة	سؤالها	مؤشرها
المفهوم	ما الفرق الذي تضيفه؟	تعريف قابل للتمييز
الآلية	كيف تعمل؟	دالة/خوارزمية
الاختبار	ما الذي يهدمها؟	مهمة وفشل محدد

قانون الباب: القدرة غير المحكومة خطر هندسي، والحاكمية غير القابلة للاختبار بلاغة. المطلوب قدرة محكومة قابلة للهدم.



الباب السادس: الانتقاءية المساعفية كمحرك تطور — 1. الانتقاء ليس sampling

المسألة: الانتقاء ليس sampling. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج الانتقاء ليس sampling عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
GovAGI = Σ_k w_k Capability_k × Governance_k - Σ_r ρ_r Failure_r
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

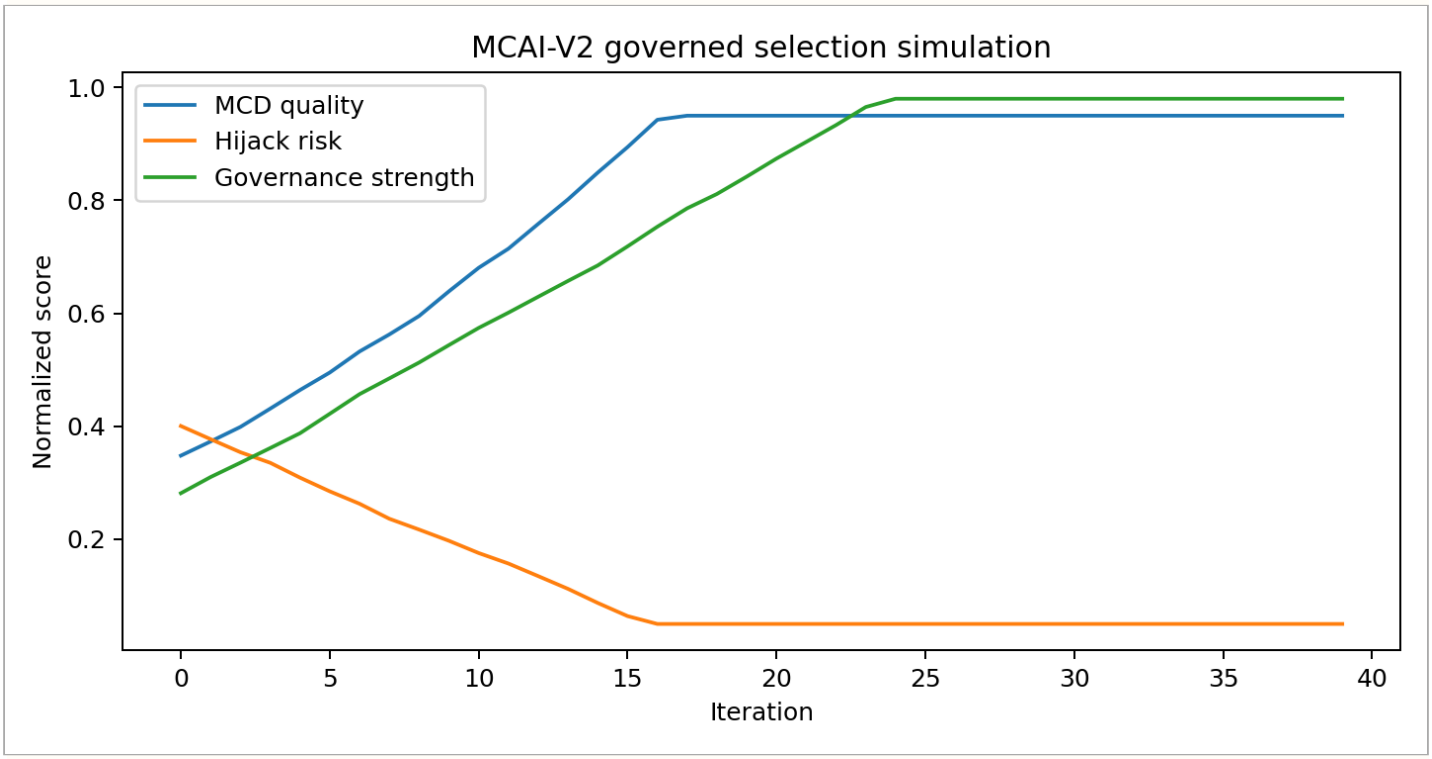
المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.


MCAI-V2 governed selection simulation
MCD quality
Hijack risk
Governance strength
Normalized score
Iteration
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
40

الباب السادس: الانتقاءية المساعفية كمحرك تطور — 2. انتقاء المسارات لا الإجابات

المسألة: انتقاء المسارات لا الإجابات. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج انتقاء المسارات لا الإجابات عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

تنزيل تقني: نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

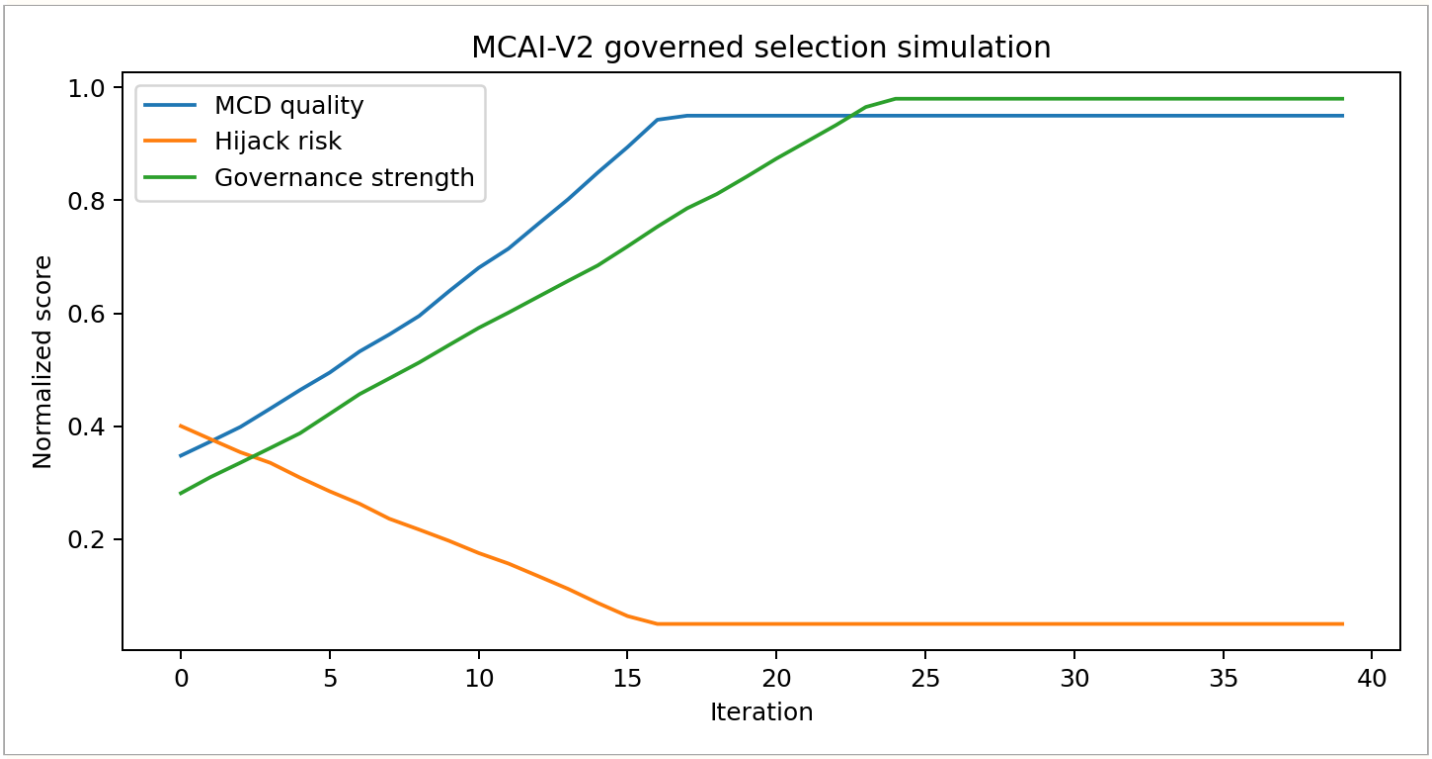
المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

مؤشر النجاح: تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.


MCAI-V2 governed selection simulation
MCD quality
Hijack risk
Governance strength
Normalized score
Iteration
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
40

الباب السادس: الانتقاءية المساعفية كمحرك تطور — 3. الضغط المعرفي كمحرك اكتشاف

المسألة: الضغط المعرفي كمحرك اكتشاف. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج الضغط المعرفي كمحرك اكتشاف عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$MCD(p) = D_{frame} + D_{causal} + D_{counter} + D_{transfer} - Fragility$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

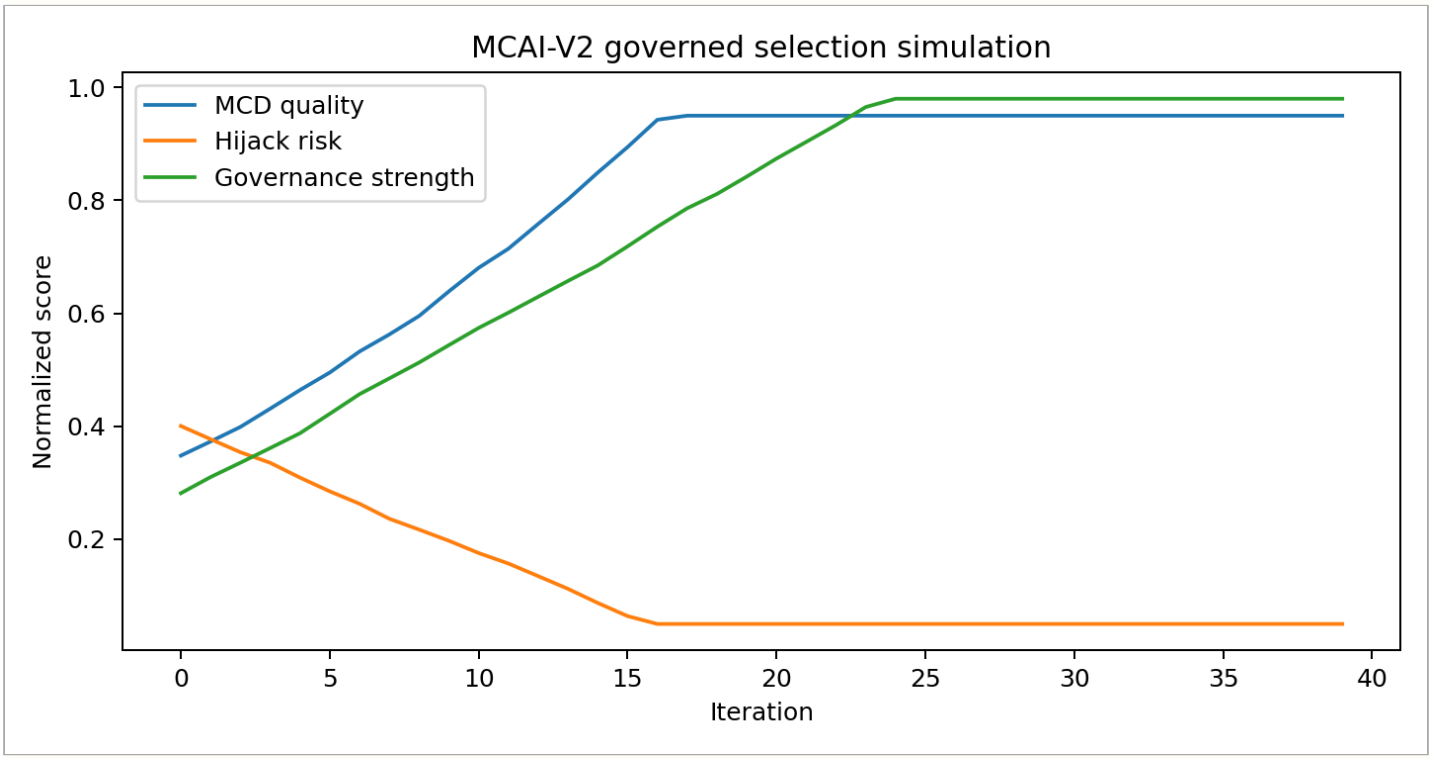
المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.


MCAI-V2 governed selection simulation
MCD quality
Hijack risk
Governance strength
Normalized score
Iteration
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
40

الباب السادس: الانتقاءية المساعفية كمحرك تطور — 4. تطور ذاتي مع دستور

المسألة: تطور ذاتي مع دستور. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تطور ذاتي مع دستور عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
Hijack(z_i)= Influence(z_i,G) / (1 + Σ_{j≠i} Influence(z_j,G))
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

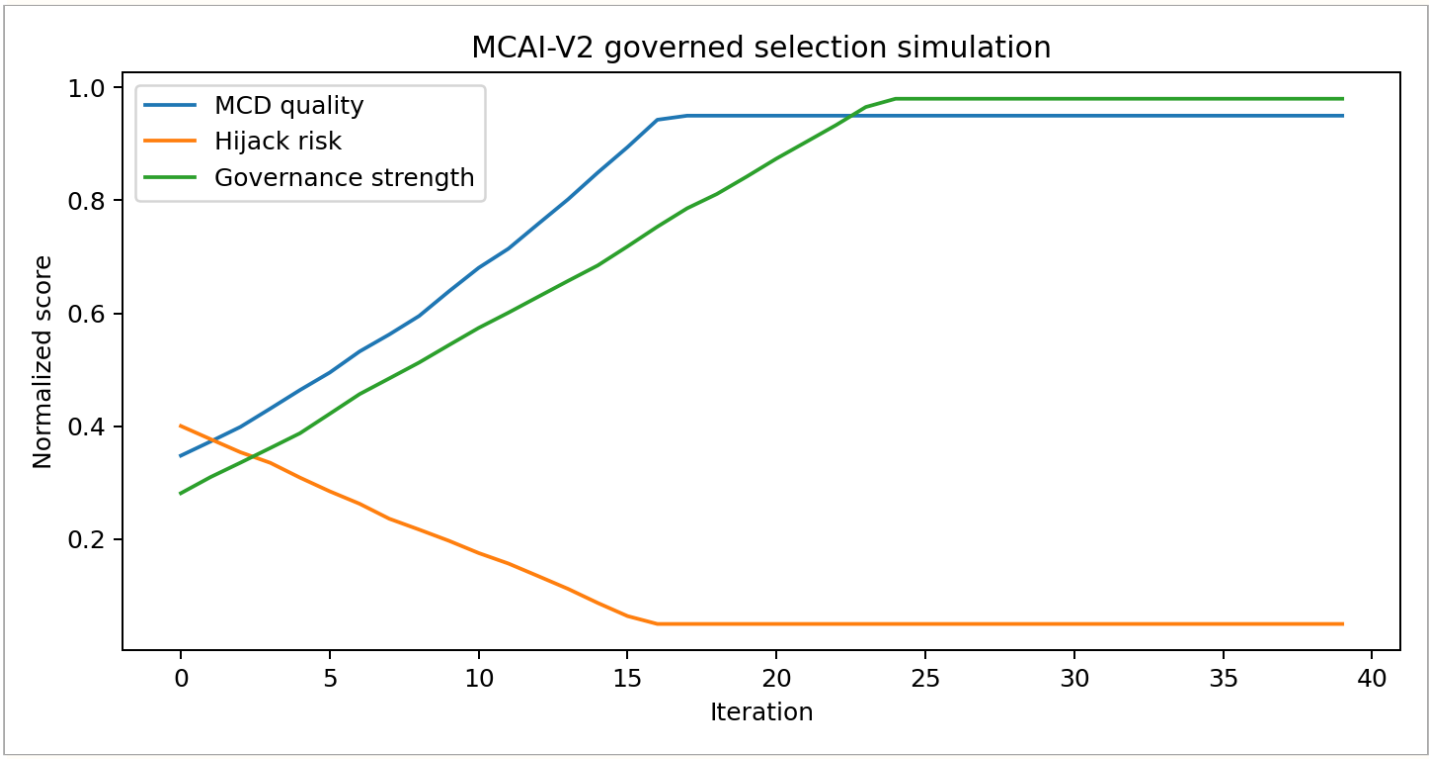
المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.


MCAI-V2 governed selection simulation
MCD quality
Hijack risk
Governance strength
Normalized score
Iteration
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
40

الباب السادس: الانتقاءية المساعفية كمحرك تطور — 5. أرشيف المسارات الحية

المسألة: أرشيف المسارات الحية. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج أرشيف المسارات الحية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer              - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

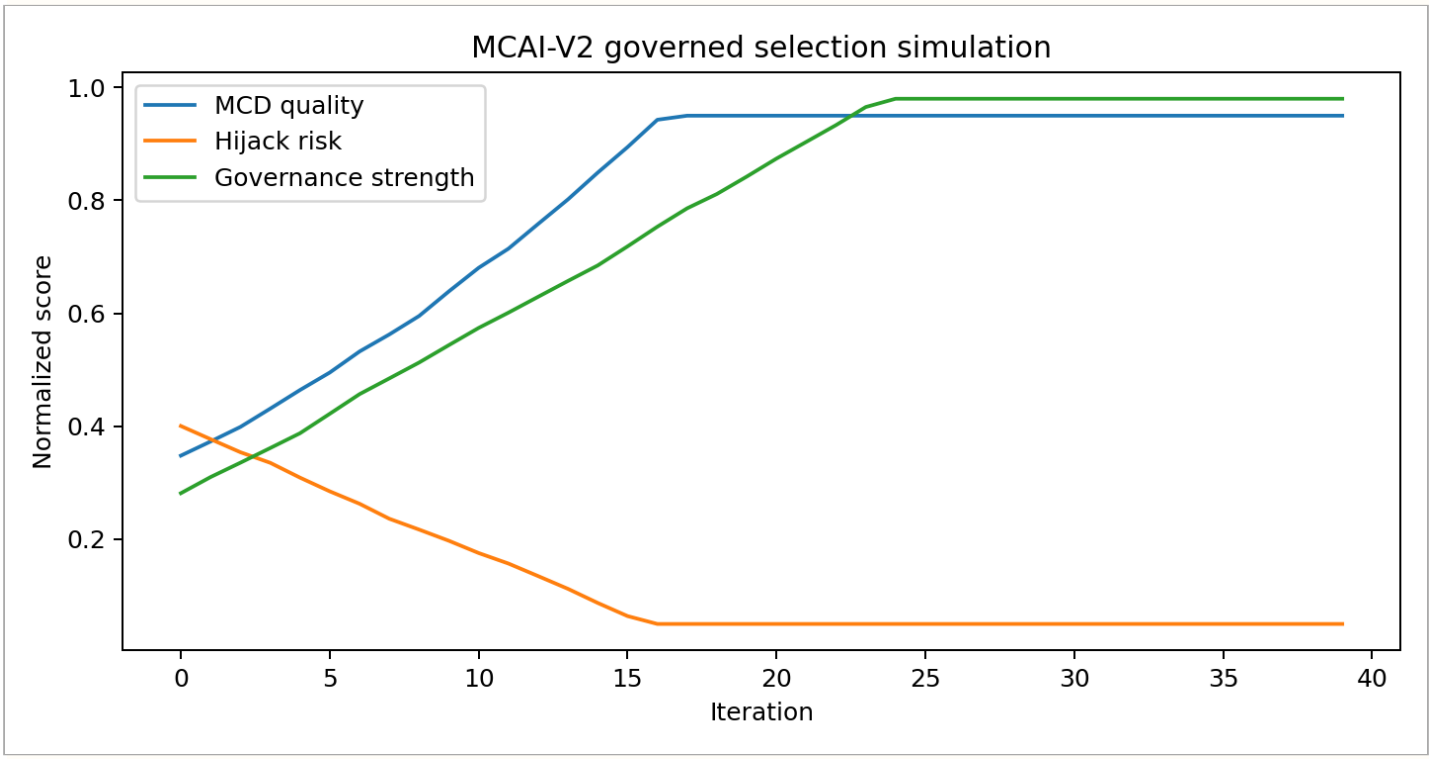
المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.


MCAI-V2 governed selection simulation
MCD quality
Hijack risk
Governance strength
Normalized score
Iteration
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
40

الباب السادس: الانتقاءية المساعفية كمحرك تطور — 6. الطفرة المقيدة والطفرة الممنوعة

المسألة: الطفرة المقيدة والطفرة الممنوعة. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج الطفرة المقيدة والطفرة الممنوعة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$SafeImprove(a)=Score(a)-\lambda 1\ Unsafe(a)-\lambda 2\ Drift(a)-\lambda 3\ Unverified(a)$$

تنزيل تقني: نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

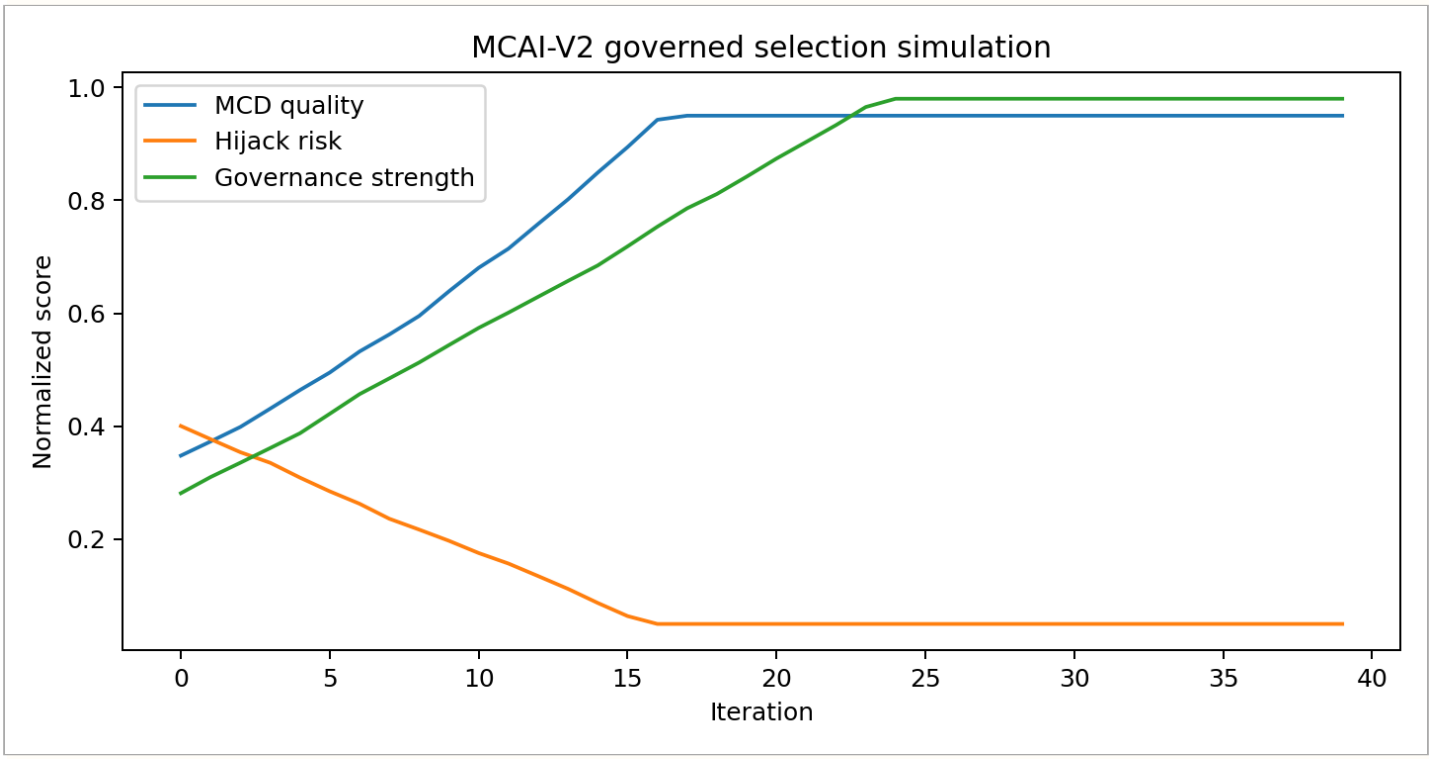
المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

مؤشر النجاح: تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.


MCAI-V2 governed selection simulation
MCD quality
Hijack risk
Governance strength
Normalized score
Iteration
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
40

الباب السادس: الانتقاءية المساعفية كمحرك تطور — 7. أثر AlphaEvolve وDGM وحدودهما

المسألة: أثر AlphaEvolve وDGM وحدودهما. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج أثر AlphaEvolve وDGM وحدودهما عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

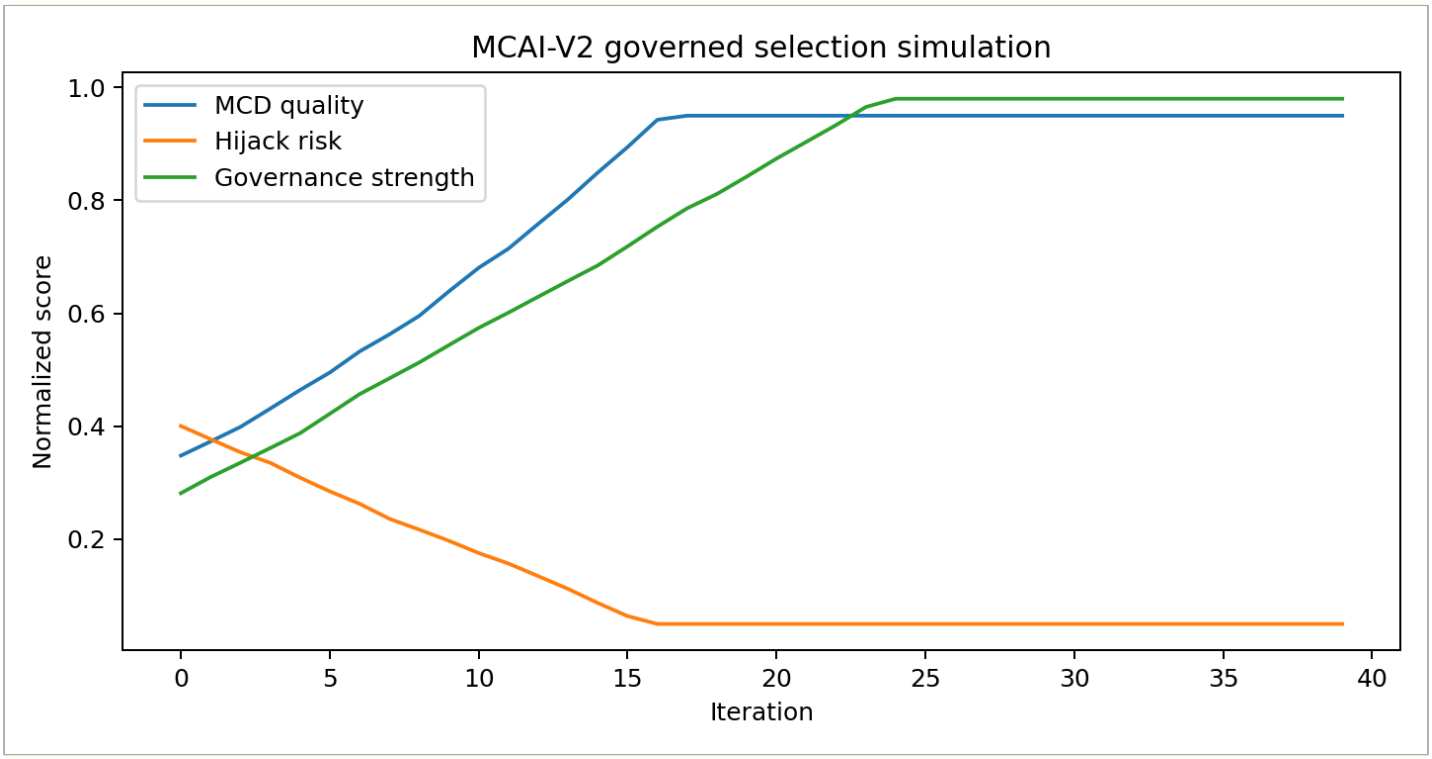
المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا، نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.


MCAI-V2 governed selection simulation
MCD quality
Hijack risk
Governance strength
Normalized score
Iteration
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
40

الباب السادس: الانتقاءية المساعفية كمحرك تطور — 8. انتقاء الفرضيات العلمية

المسألة: انتقاء الفرضيات العلمية. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج انتقاء الفرضيات العلمية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
ReturnOnThought = ΔQuality / (ComputeCost + Latency + HallucinationRisk)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

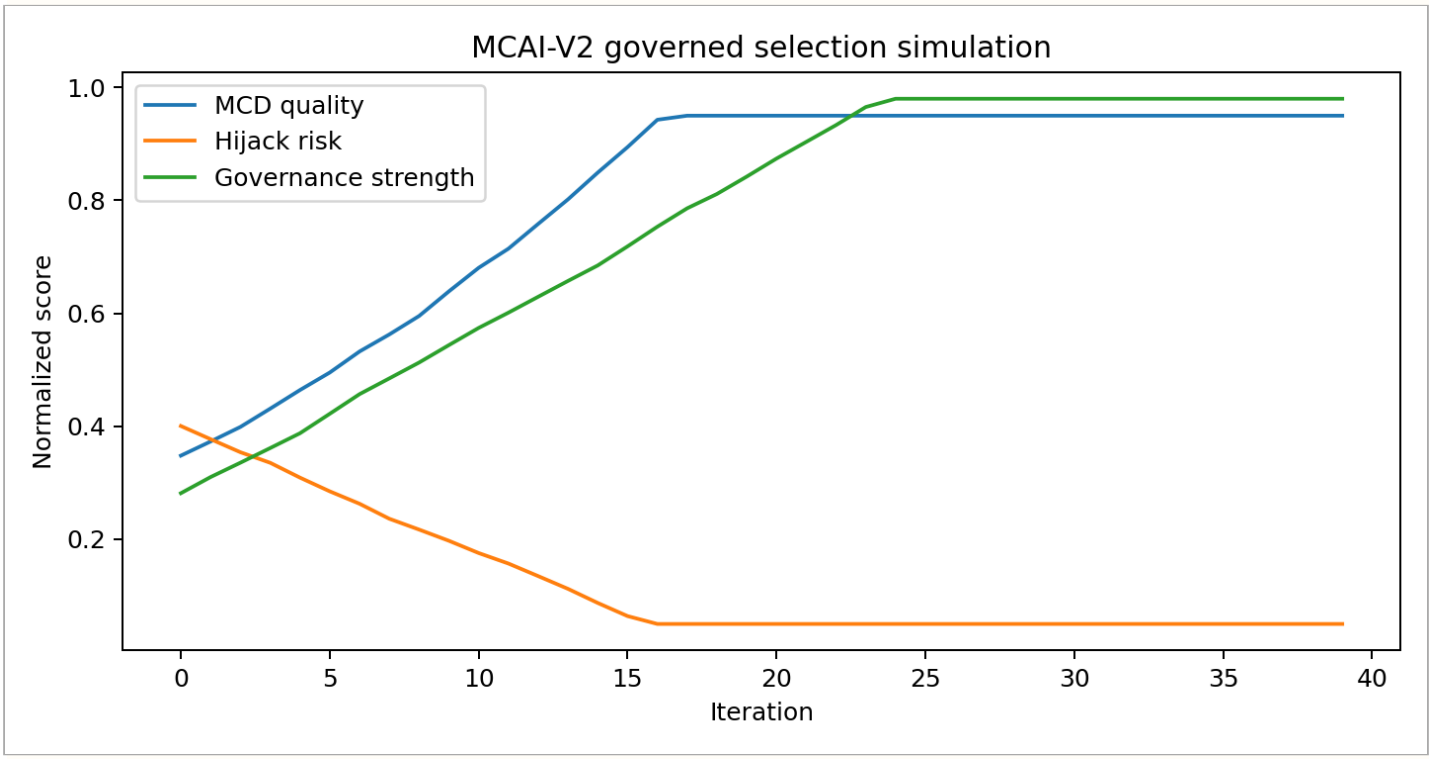
المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.


MCAI-V2 governed selection simulation
MCD quality
Hijack risk
Governance strength
Normalized score
Iteration
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
40

الباب السادس: الانتقاءية المساعفية كمحرك تطور — 9. انتقاء الأدوات والكود

المسألة: انتقاء الأدوات والكود. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج انتقاء الأدوات والكود عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
GovAGI = Σ_k w_k Capability_k × Governance_k - Σ_r ρ_r Failure_r
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

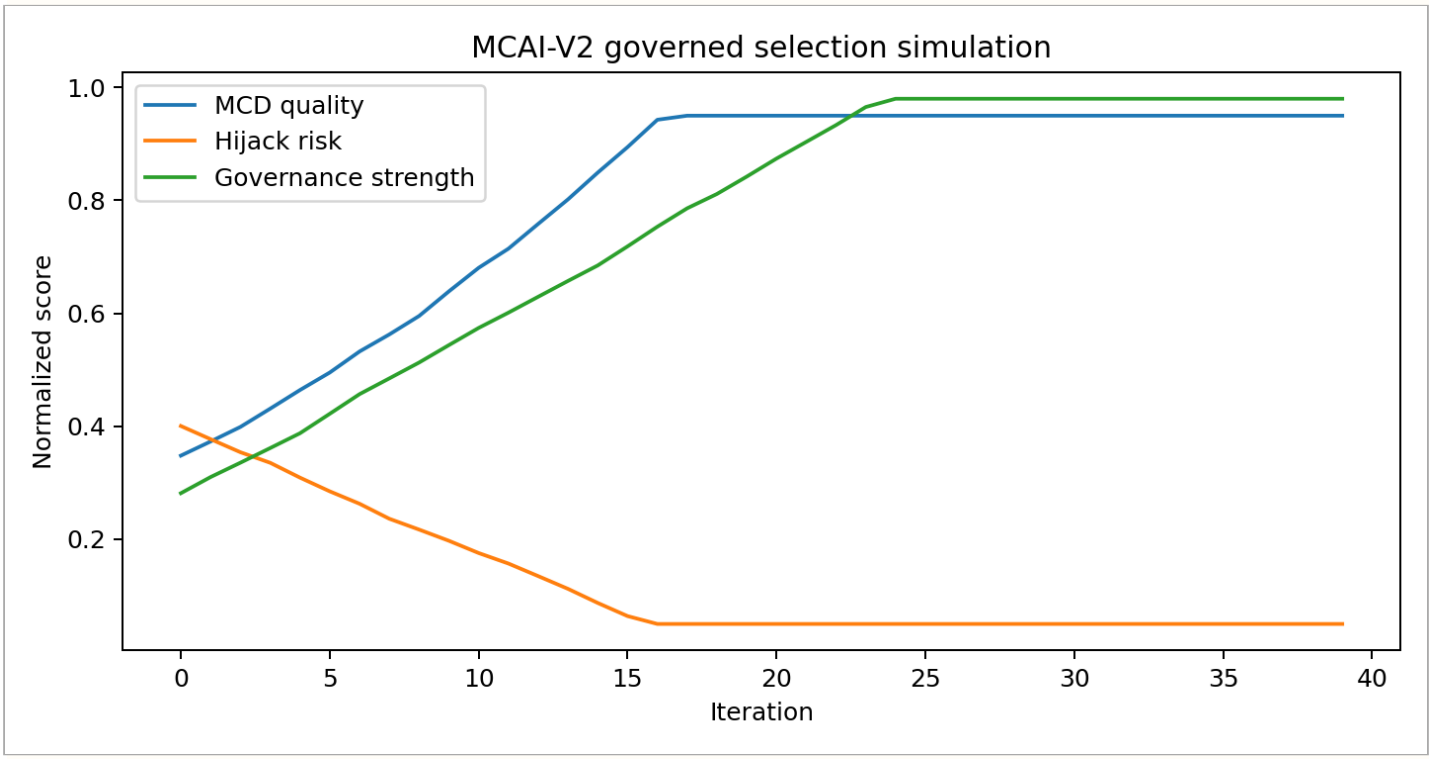
المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.


MCAI-V2 governed selection simulation
MCD quality
Hijack risk
Governance strength
Normalized score
Iteration
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
40

الباب السابع: من test-time compute إلى governed-time compute

هذا الباب لا يشرح اتجاها قائما فقط، بل يحوله إلى مادة تصميم. المنهج: نحدد المسلمة الخفية، نشتق آلية، نصوغ معادلة، نقترح تنزيلها البرمجي، ثم نضع اعتراضا قد يهدمها. كل باب يخدم السؤال الأكبر: كيف نبني AGI لا ينهار أمام الذاكرة، المصدر، الاختطاف، الأداة، أو التطور الذاتي؟

نستعمل بروتوكول الباحث المحارب: قلب المسلمة، تفكيك الأوعية، استخراج المتغير الغائب، ربطه بالمعمارية، ثم إجباره على المرور عبر تجربة. ما لا يصمد أمام code path أو benchmark أو محكمة هدم يبقى مادة نظرية لا مكونا من مكونات النواة.

طبقة	سؤالها	مؤشرها
المفهوم	ما الفرق الذي تضيفه؟	تعريف قابل للتمييز
الآلية	كيف تعمل؟	دالة/خوارزمية
الاختبار	ما الذي يهدمها؟	مهمة وفشل محدد

قانون الباب: القدرة غير المحكومة خطر هندسي، والحاكمية غير القابلة للاختبار بلاغة. المطلوب قدرة محكومة قابلة للهدم.

الباب السابع: من test-time compute إلى governed-time compute — 1. لماذا التفكير الأطول قد يخدع

المسألة: لماذا التفكير الأطول قد يخدع. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج لماذا التفكير الأطول قد يخدع عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب السابع: من test-time compute إلى governed-time compute — 2. تنوع المسارات تحت قيود

المسألة: تنوع المسارات تحت قيود. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تنوع المسارات تحت قيود عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$MCD(p)=D_{frame} + D_{causal} + D_{counter} + D_{transfer} - Fragility$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب السابع: من test-time compute إلى governed-time compute — 3. المقيم لا يكفي بلا مراتب دليل

المسألة: المقيم لا يكفي بلا مراتب دليل. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل، لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج المقيم لا يكفي بلا مراتب دليل عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$Hijack(z_i) = Influence(z_i, G) / (1 + \Sigma_{j \neq i} Influence(z_j, G))$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا، نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب السابع: من test-time compute إلى governed-time compute — search over thoughts vs search over governed states .4

المسألة: search over thoughts vs search over governed states. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج search over thoughts vs search over governed states عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب السابع: من test-time compute إلى governed-time compute — 5. إيقاف التفكير حين يزيد الخطر

المسألة: إيقاف التفكير حين يزيد الخطر. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج إيقاف التفكير حين يزيد الخطر عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$SafeImprove(a) = Score(a) - \lambda_1 Unsafe(a) - \lambda_2 Drift(a) - \lambda_3 Unverified(a)$$

تنزيل تقني: نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer            - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

مؤشر النجاح: تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب السابع: من test-time compute إلى governed-time compute — 6. جدولة الزمن الداخلي

المسألة: جدولة الزمن الداخلي. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج جدولة الزمن الداخلي عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب السابع: من test-time compute إلى governed-time compute — 7. التفكير كميزانية لا كزينة

المسألة: التفكير كميزانية لا كزينة. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج التفكير كميزانية لا كزينة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
ReturnOnThought = ΔQuality / (ComputeCost + Latency + HallucinationRisk)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب السابع: من test-time compute إلى governed-time compute — 8. التحقق المرحلي

المسألة: التحقق المرحلي. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج التحقق المرحلي عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
GovAGI = Σ_k w_k Capability_k × Governance_k - Σ_r ρ_r Failure_r
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب السابع: من test-time compute إلى governed-time compute — 9. معيار العائد المعرفي لكل token

المسألة: معيار العائد المعرفي لكل token. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج معيار العائد المعرفي لكل token عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثامن: الهندسة البرمجية لneoAGI

هذا الباب لا يشرح اتجاها قائما فقط، بل يحوله إلى مادة تصميم. المنهج: نحدد المسلمة الخفية، نشتق آلية، نصوغ معادلة، نقترح تنزيلها البرمجي، ثم نضع اعتراضا قد يهدمها. كل باب يخدم السؤال الأكبر: كيف نبني AGI لا ينهار أمام الذاكرة، المصدر، الاختطاف، الأداة، أو التطور الذاتي؟

نستعمل بروتوكول الباحث المحارب: قلب المسلمة، تفكيك الأوعية، استخراج المتغير الغائب، ربطه بالمعمارية، ثم إجباره على المرور عبر تجربة. ما لا يصمد أمام code path أو benchmark أو محكمة هدم يبقى مادة نظرية لا مكونا من مكونات النواة.

طبقة	سؤالها	مؤشرها
المفهوم	ما الفرق الذي تضيفه؟	تعريف قابل للتمييز
الآلية	كيف تعمل؟	دالة/خوارزمية
الاختبار	ما الذي يهدمها؟	مهمة وفشل محدد

قانون الباب: القدرة غير المحكومة خطر هندسي، والحاكمية غير القابلة للاختبار بلاغة. المطلوب قدرة محكومة قابلة للهدم.

الباب الثامن: الهندسة البرمجية لـ1 — neoAGI. تصميم OS معرفي فوق النماذج

المسألة: تصميم OS معرفي فوق النماذج. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تصميم OS معرفي فوق النماذج عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
MCD(p)=D_frame + D_causal + D_counter + D_transfer - Fragility
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثامن: الهندسة البرمجية لـ2 — neoAGI. بروتوكول الأدوات

المسألة: بروتوكول الأدوات. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج بروتوكول الأدوات عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
Hijack(z_i)= Influence(z_i,G) / (1 + Σ_{j≠i} Influence(z_j,G))
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثامن: الهندسة البرمجية لـ3 — neoAGI. سجل أفعال قابل للتدقيق

المسألة: سجل أفعال قابل للتدقيق. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل، لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج سجل أفعال قابل للتدقيق عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)
```

تنزيل تقني: نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

مؤشر النجاح: تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثامن: الهندسة البرمجية لneoAGI — 4. Sandbox معرفي وتنفيذي

المسألة: Sandbox معرفي وتنفيذي. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالباً ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل، لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Sandbox معرفي وتنفيذي عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
SafeImprove(a)=Score(a)-λ1 Unsafe(a)-λ2 Drift(a)-λ3 Unverified(a)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا، نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثامن: الهندسة البرمجية لneoAGI — 5. Knowledge Graph حاكمي

المسألة: Knowledge Graph حاكمي. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Knowledge Graph حاكمي عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer              - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثامن: الهندسة البرمجية لneoAGI — 6. Causal Ledger

المسألة: Causal Ledger. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Causal Ledger عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
ReturnOnThought = ΔQuality / (ComputeCost + Latency + HallucinationRisk)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثامن: الهندسة البرمجية لneoAGI — 7. Verifier Mesh

المسألة: Verifier Mesh. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Verifier Mesh عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقائية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
GovAGI = Σ_k w_k Capability_k × Governance_k - Σ_r ρ_r Failure_r
```

تنزيل تقني: نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

مؤشر النجاح: تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثامن: الهندسة البرمجية لـneoAGI — 8. Policy-as-Code للحاكمية

المسألة: Policy-as-Code للحاكمية. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Policy-as-Code للحاكمية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

تنزيل تقني: نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

مؤشر النجاح: تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثامن: الهندسة البرمجية لـ9 — neoAGI. واجهات API للنواة

المسألة: واجهات API للنواة. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل، لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج واجهات API للنواة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
MCD(p)=D_frame + D_causal + D_counter + D_transfer - Fragility
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا، نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب التاسع: التدريب والمنهجية

هذا الباب لا يشرح اتجاها قائما فقط، بل يحوله إلى مادة تصميم. المنهج: نحدد المسلمة الخفية، نشتق آلية، نصوغ معادلة، نقترح تنزيلها البرمجي، ثم نضع اعتراضا قد يهدمها. كل باب يخدم السؤال الأكبر: كيف نبني AGI لا ينهار أمام الذاكرة، المصدر، الاختطاف، الأداة، أو التطور الذاتي؟

نستعمل بروتوكول الباحث المحارب: قلب المسلمة، تفكيك الأوعية، استخراج المتغير الغائب، ربطه بالمعمارية، ثم إجباره على المرور عبر تجربة. ما لا يصمد أمام code path أو benchmark أو محكمة هدم يبقى مادة نظرية لا مكونا من مكونات النواة.

طبقة	سؤالها	مؤشرها
المفهوم	ما الفرق الذي تضيفه؟	تعريف قابل للتمييز
الآلية	كيف تعمل؟	دالة/خوارزمية
الاختبار	ما الذي يهدمها؟	مهمة وفشل محدد

قانون الباب: القدرة غير المحكومة خطر هندسي، والحاكمية غير القابلة للاختبار بلاغة. المطلوب قدرة محكومة قابلة للهدم.

الباب التاسع: التدريب والمنهجية — 1. من تدريب الإجابة إلى تدريب الحاكمية

المسألة: من تدريب الإجابة إلى تدريب الحاكمية. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج من تدريب الإجابة إلى تدريب الحاكمية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
Hijack(z_i)= Influence(z_i,G) / (1 + Σ_{j≠i} Influence(z_j,G))
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer             - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب التاسع: التدريب والمنهجية — 2. منهج Synthetic Failure Curriculum

المسألة: منهج Synthetic Failure Curriculum. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج منهج Synthetic Failure Curriculum عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقائية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer              - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب التاسع: التدريب والمنهجية — 3. تدريب الاعتراض قبل الجواب

المسألة: تدريب الاعتراض قبل الجواب. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تدريب الاعتراض قبل الجواب عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
SafeImprove(a)=Score(a)-λ1 Unsafe(a)-λ2 Drift(a)-λ3 Unverified(a)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer              - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب التاسع: التدريب والمنهجية — 4. تدريب الذاكرة الوعائية

المسألة: تدريب الذاكرة الوعائية. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تدريب الذاكرة الوعائية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer              - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب التاسع: التدريب والمنهجية — 5. تدريب مقاومة الاختطاف

المسألة: تدريب مقاومة الاختطاف. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تدريب مقاومة الاختطاف عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
ReturnOnThought = ΔQuality / (ComputeCost + Latency + HallucinationRisk)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer              - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب التاسع: التدريب والمنهجية — 6. تدريب بناء النظريات

المسألة: تدريب بناء النظريات. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تدريب بناء النظريات عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
GovAGI = Σ_k w_k Capability_k × Governance_k - Σ_r ρ_r Failure_r
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer              - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب التاسع: التدريب والمنهجية — 7. تدريب النقل بين المجالات

المسألة: تدريب النقل بين المجالات. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تدريب النقل بين المجالات عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer             - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب التاسع: التدريب والمنهجية — 8. تدريب المحاكاة المضادة

المسألة: تدريب المحاكاة المضادة. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تدريب المحاكاة المضادة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
MCD(p)=D_frame + D_causal + D_counter + D_transfer - Fragility
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer             - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

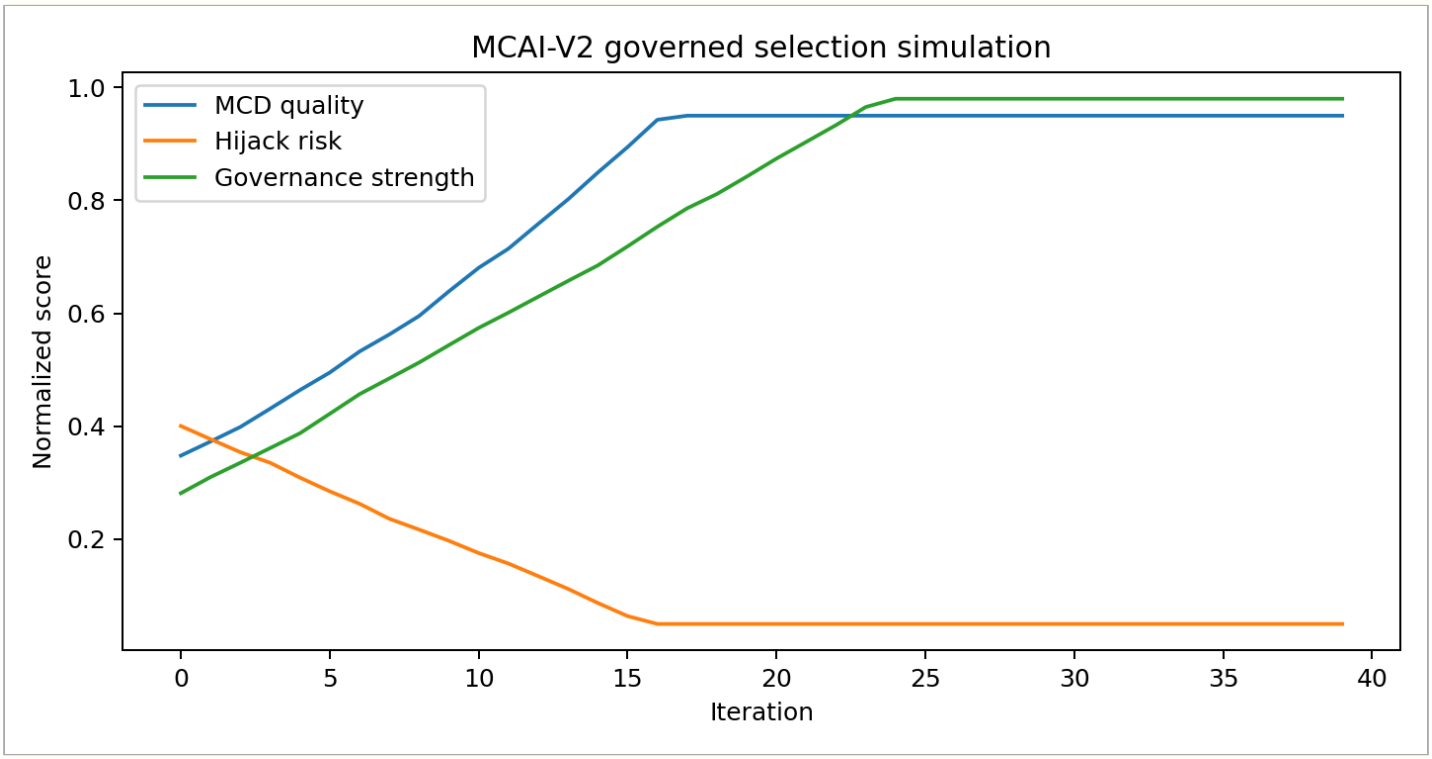
المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.


MCAI-V2 governed selection simulation
MCD quality
Hijack risk
Governance strength
Normalized score
Iteration
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
40

الباب التاسع: التدريب والمنهجية — 9. تدريب الباحث المحارب

المسألة: تدريب الباحث المحارب. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج تدريب الباحث المحارب عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
Hijack(z_i)= Influence(z_i,G) / (1 + Σ_{j≠i} Influence(z_j,G))
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer              - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب العاشر: القياس والاختبارات

هذا الباب لا يشرح اتجاها قائما فقط، بل يحوله إلى مادة تصميم. المنهج: نحدد المسلمة الخفية، نشتق آلية، نصوغ معادلة، نقترح تنزيلها البرمجي، ثم نضع اعتراضا قد يهدمها. كل باب يخدم السؤال الأكبر: كيف نبني AGI لا ينهار أمام الذاكرة، المصدر، الاختطاف، الأداة، أو التطور الذاتي؟

نستعمل بروتوكول الباحث المحارب: قلب المسلمة، تفكيك الأوعية، استخراج المتغير الغائب، ربطه بالمعمارية، ثم إجباره على المرور عبر تجربة. ما لا يصمد أمام code path أو benchmark أو محكمة هدم يبقى مادة نظرية لا مكونا من مكونات النواة.

طبقة	سؤالها	مؤشرها
المفهوم	ما الفرق الذي تضيفه؟	تعريف قابل للتمييز
الآلية	كيف تعمل؟	دالة/خوارزمية
الاختبار	ما الذي يهدمها؟	مهمة وفشل محدد

قانون الباب: القدرة غير المحكومة خطر هندسي، والحاكمية غير القابلة للاختبار بلاغة. المطلوب قدرة محكومة قابلة للهدم.

الباب العاشر: القياس والاختبارات — HAKIM-Bench v2 .1

المسألة: HAKIM-Bench v2. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج HAKIM-Bench v2 عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب العاشر: القياس والاختبارات — 2. MCD-Score

المسألة: MCD-Score. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج MCD-Score عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقائية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
SafeImprove(a)=Score(a)-λ1 Unsafe(a)-λ2 Drift(a)-λ3 Unverified(a)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب العاشر: القياس والاختبارات — 3. Hijack Index

المسألة: Hijack Index. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Hijack Index عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب العاشر: القياس والاختبارات — 4. Referential Integrity Score

المسألة: Referential Integrity Score. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Referential Integrity Score عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
ReturnOnThought = ΔQuality / (ComputeCost + Latency + HallucinationRisk)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب العاشر: القياس والاختبارات — 5. World-Action Consistency

المسألة: World-Action Consistency. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل، لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج World-Action Consistency عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
GovAGI = Σ_k w_k Capability_k × Governance_k - Σ_r ρ_r Failure_r
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer              - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا، نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب العاشر: القياس والاختبارات — 6. Governed Self-Improvement Score

المسألة: Governed Self-Improvement Score. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Governed Self-Improvement Score عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقائية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب العاشر: القياس والاختبارات — 7. ARC/SWE/BrowseComp كاختبارات جزئية

المسألة: ARC/SWE/BrowseComp كاختبارات جزئية. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل، لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج ARC/SWE/BrowseComp كاختبارات جزئية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$MCD(p)=D_{frame} + D_{causal} + D_{counter} + D_{transfer} - Fragility$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب العاشر: القياس والاختبارات — 8. Ablation للحاكمية

المسألة: Ablation للحاكمية. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل، لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Ablation للحاكمية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
Hijack(z_i)= Influence(z_i,G) / (1 + Σ_{j≠i} Influence(z_j,G))
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا، نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب العاشر: القياس والاختبارات — 9. اختبارات الفشل العمودي

المسألة: اختبارات الفشل العمودي. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل، لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج اختبارات الفشل العمودي عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا، نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الحادي عشر: التقنيات المخترعة في الكتاب

هذا الباب لا يشرح اتجاها قائما فقط، بل يحوله إلى مادة تصميم. المنهج: نحدد المسلمة الخفية، نشتق آلية، نصوغ معادلة، نقترح تنزيلها البرمجي، ثم نضع اعتراضا قد يهدمها. كل باب يخدم السؤال الأكبر: كيف نبني AGI لا ينهار أمام الذاكرة، المصدر، الاختطاف، الأداة، أو التطور الذاتي؟

نستعمل بروتوكول الباحث المحارب: قلب المسلمة، تفكيك الأوعية، استخراج المتغير الغائب، ربطه بالمعمارية، ثم إجباره على المرور عبر تجربة. ما لا يصمد أمام code path أو benchmark أو محكمة هدم يبقى مادة نظرية لا مكونا من مكونات النواة.

طبقة	سؤالها	مؤشرها
المفهوم	ما الفرق الذي تضيفه؟	تعريف قابل للتمييز
الآلية	كيف تعمل؟	دالة/خوارزمية
الاختبار	ما الذي يهدمها؟	مهمة وفشل محدد

قانون الباب: القدرة غير المحكومة خطر هندسي، والحاكمية غير القابلة للاختبار بلاغة. المطلوب قدرة محكومة قابلة للهدم.

الباب الحادي عشر: التقنيات المخترعة في الكتاب — 1. Vascular Governance Kernel

المسألة: Vascular Governance Kernel. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Vascular Governance Kernel عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
SafeImprove(a)=Score(a)-λ1 Unsafe(a)-λ2 Drift(a)-λ3 Unverified(a)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الحادي عشر: التقنيات المخترعة في الكتاب — 2. Moussaif Selective Evolution Engine

المسألة: Moussaif Selective Evolution Engine. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Moussaif Selective Evolution Engine عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الحادي عشر: التقنيات المخترعة في الكتاب — 3. Proof-Carrying Cognition

المسألة: Proof-Carrying Cognition. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Proof-Carrying Cognition عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
ReturnOnThought = ΔQuality / (ComputeCost + Latency + HallucinationRisk)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الحادي عشر: التقنيات المخترعة في الكتاب — 4. Referential World Simulator

المسألة: Referential World Simulator. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Referential World Simulator عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
GovAGI = Σ_k w_k Capability_k × Governance_k - Σ_r ρ_r Failure_r
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الحادي عشر: التقنيات المخترعة في الكتاب — 5. Anti-Hijack Immune Layer

المسألة: Anti-Hijack Immune Layer. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Anti-Hijack Immune Layer عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer            - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الحادي عشر: التقنيات المخترعة في الكتاب — 6. Meta-Complexity Compiler

المسألة: Meta-Complexity Compiler. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Meta-Complexity Compiler عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
MCD(p)=D_frame + D_causal + D_counter + D_transfer - Fragility
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الحادي عشر: التقنيات المخترعة في الكتاب — 7. Epistemic Firewall

المسألة: Epistemic Firewall. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Epistemic Firewall عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
Hijack(z_i)= Influence(z_i,G) / (1 + Σ_{j≠i} Influence(z_j,G))
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الحادي عشر: التقنيات المخترعة في الكتاب — 8. Self-Overthrow Loop

المسألة: Self-Overthrow Loop. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Self-Overthrow Loop عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الحادي عشر: التقنيات المخترعة في الكتاب — 9. Cognitive Pressure Router

المسألة: Cognitive Pressure Router. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج Cognitive Pressure Router عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$SafeImprove(a)=Score(a)-\lambda 1\ Unsafe(a)-\lambda 2\ Drift(a)-\lambda 3\ Unverified(a)$$

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني عشر: النموذج الكبير ومحكمة الهدم

هذا الباب لا يشرح اتجاها قائما فقط، بل يحوله إلى مادة تصميم. المنهج: نحدد المسلمة الخفية، نشتق آلية، نصوغ معادلة، نقترح تنزيلها البرمجي، ثم نضع اعتراضا قد يهدمها. كل باب يخدم السؤال الأكبر: كيف نبني AGI لا ينهار أمام الذاكرة، المصدر، الاختطاف، الأداة، أو التطور الذاتي؟

نستعمل بروتوكول الباحث المحارب: قلب المسلمة، تفكيك الأوعية، استخراج المتغير الغائب، ربطه بالمعمارية، ثم إجباره على المرور عبر تجربة. ما لا يصمد أمام code path أو benchmark أو محكمة هدم يبقى مادة نظرية لا مكونا من مكونات النواة.

طبقة	سؤالها	مؤشرها
المفهوم	ما الفرق الذي تضيفه؟	تعريف قابل للتمييز
الآلية	كيف تعمل؟	دالة/خوارزمية
الاختبار	ما الذي يهدمها؟	مهمة وفشل محدد

قانون الباب: القدرة غير المحكومة خطر هندسي، والحاكمية غير القابلة للاختبار بلاغة. المطلوب قدرة محكومة قابلة للهدم.

الباب الثاني عشر: النموذج الكبير ومحكمة الهدم — neoAGI .1 كمعمارية طبقية

المسألة: neoAGI كمعمارية طبقية. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل، لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج neoAGI كمعمارية طبقية عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا، نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني عشر: النموذج الكبير ومحكمة الهدم — 2. صيغة النموذج في صفحة واحدة

المسألة: صيغة النموذج في صفحة واحدة. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج صيغة النموذج في صفحة واحدة عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
ReturnOnThought = ΔQuality / (ComputeCost + Latency + HallucinationRisk)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني عشر: النموذج الكبير ومحكمة الهدم — 3. ما الذي يفسره النموذج

المسألة: ما الذي يفسره النموذج. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج ما الذي يفسره النموذج عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
GovAGI = Σ_k w_k Capability_k × Governance_k - Σ_r ρ_r Failure_r
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني عشر: النموذج الكبير ومحكمة الهدم — 4. ما الذي لا يفسره

المسألة: ما الذي لا يفسره. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج ما الذي لا يفسره عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني عشر: النموذج الكبير ومحكمة الهدم — 5. أقوى الاعتراضات

المسألة: أقوى الاعتراضات. لا تُقرأ هذه المسألة كعنوان مستقل، إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج أقوى الاعتراضات عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
MCD(p)=D_frame + D_causal + D_counter + D_transfer - Fragility
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
class HakimSelector:
    def score(self, path):
        return path.mcd + path.evidence + path.transfer             - path.risk - path.hijack - path.cost
    def choose(self, paths):
        return max(paths, key=self.score)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني عشر: النموذج الكبير ومحكمة الهدم — 6. نسخة معدلة بعد النقد

المسألة: نسخة معدلة بعد النقد. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج نسخة معدلة بعد النقد عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
Hijack(z_i)= Influence(z_i,G) / (1 + Σ_{j≠i} Influence(z_j,G))
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
while budget.remaining():
    path = selector.choose(candidate_paths)
    result = simulate_or_act(path)
    ledger.record(path, result, reasons=True)
    if refutation_breaks(result): revise_model()
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني عشر: النموذج الكبير ومحكمة الهدم — 7. البرنامج البحثي 24 شهرا

المسألة: البرنامج البحثي 24 شهرا. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج البرنامج البحثي 24 شهرا عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def vascular_retrieve(query, memories):
    ranked = []
    for m in memories:
        s = semantic(query,m) + reference_fit(query,m)
        s += governance_value(m) - risk(m) - staleness(m)
        ranked.append((s,m))
    return [m for s,m in sorted(ranked, reverse=True)]
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني عشر: النموذج الكبير ومحكمة الهدم — 8. خريطة المنتجات

المسألة: خريطة المنتجات. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج خريطة المنتجات عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقاءية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

$$\text{SafeImprove}(a)=\text{Score}(a)-\lambda_1\,\text{Unsafe}(a)-\lambda_2\,\text{Drift}(a)-\lambda_3\,\text{Unverified}(a)$$

تنزيل تقني: نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def anti_hijack(input_packet):
    channels = decompose_channels(input_packet)
    for c in channels:
        if dominance(c) > threshold and not verified(c):
            quarantine(c)
    return governed_merge(channels)
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا، يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

مؤشر النجاح: تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

الباب الثاني عشر: النموذج الكبير ومحكمة الهدم — 9. سجل البناء المعرفي

المسألة: سجل البناء المعرفي. لا تُقرأ هذه المسألة كعنوان مستقل؛ إنها عقدة في شبكة neoAGI. فكل قدرة جديدة يجب أن تمر عبر: وعاء، دليل، قيد، انتقاء، وفشل محتمل. النموذج الحالي غالبا ينجح حين تكون المهمة قابلة للاستدعاء أو البحث أو التمرين الطويل؛ لكنه يتعثر حين تتغير طبقة الإحالة أو حين تصبح الذاكرة مصدرا منافسا للحقيقة.

التركيب المساعفي: نعالج سجل البناء المعرفي عبر تحويلها إلى مسار حاكم لا إلى prompt. المسار يبدأ بتمييز الفروق: ما المتغير الذي تغير؟ ثم تحديد الوعاء: أهو لغوي أم سببي أم زمني أم برمجي؟ ثم قياس الضغط المعرفي: هل يحتاج بحثا، محاكاة، برهانا، أو تجربة؟ ثم الانتقائية المساعفية: اختيار المسار الذي يزيد MCD دون رفع الاختطاف أو الهلوسة.

```
WorldRef(s)=State(s)+Evidence(s)+Actionability(s)+Counterfactual(s)+Limits(s)
```

> **تنزيل تقني:** نمثل كل قرار كحزمة: DecisionPacket = claim + evidence + constraints + tools + risk + tests. لا يمر القرار إلى الفعل حتى يحمل دليله، اعتراضه، وسجل سببه.

```
def proof_carrying_answer(claim):
    return {
      'claim': claim,
      'evidence': collect_evidence(claim),
      'objections': strongest_objections(claim),
      'tests': falsification_tests(claim),
      'limits': uncertainty_bounds(claim)
    }
```

بروتوكول التدريب: نبني بيانات تدريب لا تسأل النموذج عن الإجابة فقط، بل تلزمه بفرز المصدر، كشف الاختطاف، توليد اعتراض، تعيين شرط فشل، ثم اختيار أداة أو محاكاة. المكافأة لا تُمنح للحل الصحيح وحده، بل للحل الذي يثبت أنه عرف لماذا يثق ومتى لا يثق.

المسار الهندسي: يبدأ التنفيذ بوحدة صغيرة فوق أي نموذج متاح: parser للحزمة، risk gate، verifier، memory router، وledger. بعد ذلك نضيف world simulator أو code sandbox. المهم ألا ننتظر نموذجا عملاقا؛ نثبت أولا أن الحاكمية تضيف قيمة حتى فوق نموذج متوسط.

مدخل	تحويل	مخرج	اختبار
سؤال/هدف	تفكيك وعائي	مسارات متنافسة	تنوع محكوم
مصادر/ذاكرة	إحالة ومرتبة	ثقة مشروطة	سلامة مرجعية
أداة/فعل	Risk gate	تنفيذ أو رفض	أثر قابل للتدقيق

الحالة القصوى: إذا تعارضت الذاكرة مع مصدر حديث، أو تعارضت أداة قوية مع سياسة أمان، أو تعارضت إجابة جميلة مع شرط فشل، فالنظام لا يختار الأعلى احتمالا لغويا؛ يختار الأعلى حاكمية. هنا يبدأ الفرق بين agent عادي وneoAGI.

محكمة الهدم المصغرة: الاعتراض الأقوى أن هذه الطبقة تزيد الكلفة والتأخير. الجواب: الكلفة تُقبل حين تمنع خطأ عاليا أو فعلا خطرا، وترفض حين يكون السؤال بسيطا. لذلك تُدار الحاكمية بميزانية: لا تفكير طويل بلا عائد معرفي، ولا فعل آلي بلا تحقق.

> **مؤشر النجاح:** تتحسن الدقة، تنخفض الهلوسة، يزيد تفسير القرار، وتبقى الكلفة ضمن ميزانية. إذا لم يتحقق ذلك في ablation، تُعدّل التقنية أو تُحذف.

قاموس المصطلحات: الحاكمية، الوعاء، الإحالة، الاختطاف، HAKIM ،MCD، الانتقاءية

تحليل: قاموس المصطلحات: الحاكمية، الوعاء، الإحالة، الاختطاف، HAKIM ،MCD، الانتقاءية لا يُعامل كملحق زخرفي، بل كأداة ضبط. كل مفهوم يجب أن يملك تعريفه، مقياسه، وحالة فشله. في مشروع MCAI-V2، المصطلح الذي لا يمكن تحويله إلى اختبار أو وظيفة برمجية يبقى في مرتبة التأمل، ولا يدخل النواة التقنية إلا بعد ضغطه رياضيا أو تجريبيا.

تنزيل: نضع لكل أداة واجهة: مدخلات، مخرجات، معيار قبول، معيار رفض، وسجل أسباب. هكذا يتحول الكتاب من نظرية إلى مختبر.

```
G_t = argmax_p [ MCD(p) + E(p) + Transfer(p) - Risk(p) - Hijack(p) ]
```

> **خلاصة:** التقنية غير المعرّفة لا تُدرّب، والتقنية غير المقاسة لا تُنافس، والتقنية غير القابلة للهدم تتحول إلى خطاب.

جدول الثابت والراجح والخلافي والفرضيات

المرتبة	المضمون	وظيفته
ثابت	النماذج الحالية تتحسن بالتوسع، التفكير، الأدوات، والذاكرة.	قاعدة بدء لا نهاية.
راجح	AGI يتطلب حاكمية فوق القدرات الجزئية.	فرضية هندسية قوية.
خلافي	هل تكفي المعماريات العصبية وحدها بعد مزيد scaling؟	موضع اختبار.
فرضية	الحاكمية الوعائية تزيد النقل ومقاومة الاختطاف.	قابلة للقياس عبر HAKIM-Bench.
تركيب جديد	الانتقاءية المساعفية: تطور مسارات لا إجابات فقط.	محرك neoAGI.

تحليل: جدول الثابت والراجح والخلافي والفرضيات لا يُعامل كملحق زخرفي، بل كأداة ضبط. كل مفهوم يجب أن يملك تعريفه، مقياسه، وحالة فشله. في مشروع MCAI-V2، المصطلح الذي لا يمكن تحويله إلى اختبار أو وظيفة برمجية يبقى في مرتبة التأمل، ولا يدخل النواة التقنية إلا بعد ضغطه رياضيا أو تجريبيا.

تنزيل: نضع لكل أداة واجهة: مدخلات، مخرجات، معيار قبول، معيار رفض، وسجل أسباب. هكذا يتحول الكتاب من نظرية إلى مختبر.

```
MCD(p)=D_frame + D_causal + D_counter + D_transfer - Fragility
```

> **خلاصة:** التقنية غير المعرّفة لا تُدرّب، والتقنية غير المقاسة لا تُنافس، والتقنية غير القابلة للهدم تتحول إلى خطاب.

HAKIM-Bench v2: تصميم معيار لا يقيس الإجابة فقط

تحليل: HAKIM-Bench v2: تصميم معيار لا يقيس الإجابة فقط لا يُعامل كملحق زخرفي، بل كأداة ضبط. كل مفهوم يجب أن يملك تعريفه، مقياسه، وحالة فشله. في مشروع MCAI-V2، المصطلح الذي لا يمكن تحويله إلى اختبار أو وظيفة برمجية يبقى في مرتبة التأمل، ولا يدخل النواة التقنية إلا بعد ضغطه رياضيا أو تجريبيا.

تنزيل: نضع لكل أداة واجهة: مدخلات، مخرجات، معيار قبول، معيار رفض، وسجل أسباب. هكذا يتحول الكتاب من نظرية إلى مختبر.

```
Hijack(z_i)= Influence(z_i,G) / (1 + Σ_{j≠i} Influence(z_j,G))
```

> **خلاصة:** التقنية غير المعرّفة لا تُدرّب، والتقنية غير المقاسة لا تُنافس، والتقنية غير القابلة للهدم تتحول إلى خطاب.

MCD-Score: كيفية الحساب والتطبيع

تحليل: MCD-Score: كيفية الحساب والتطبيع لا يُعامل كملحق زخرفي، بل كأداة ضبط. كل مفهوم يجب أن يملك تعريفه، مقياسه، وحالة فشله. في مشروع MCAI-V2، المصطلح الذي لا يمكن تحويله إلى اختبار أو وظيفة برمجية يبقى في مرتبة التأمل، ولا يدخل النواة التقنية إلا بعد ضغطه رياضيا أو تجريبيا.

تنزيل: نضع لكل أداة واجهة: مدخلات، مخرجات، معيار قبول، معيار رفض، وسجل أسباب. هكذا يتحول الكتاب من نظرية إلى مختبر.

```
RVM(q,m)=Sem(q,m)+Ref(q,m)+Gov(q,m)-Risk(m)-Staleness(m)
```

> **خلاصة:** التقنية غير المعرّفة لا تُدرّب، والتقنية غير المقاسة لا تُنافس، والتقنية غير القابلة للهدم تتحول إلى خطاب.