منظومة مساعف التشغيلية الموجّهة نحو الذكاء العام

HyperSkills - HyperKB-as-OS - HyperLoops

حالة الفن وبناء طبقة تشغيل معرفية ضمن نظرية مساعف

State of the Art and an AGI-Oriented Operational Layer for Moussaif Theory

المعرفة لا تصبح قدرة حتى تُربط بعقد مهارة، والمهارة لا تصبح وكالة حتى تُغلق حلقة موثّقة ومحدودة.

State of the Art as of 14 July 2026 - ورقة بحثية مركزة

إعداد وتركيب معماري: زهير مساعف / **Zouhair Moussaif**

Moussaif AGI-Oriented Theory - Research Edition 1.0

الحالة المعرفية: إطار نظري وهندسي أصلي مقترح، وليس دليلا على تحقق *AGI* أو إثباتا قانونيا للأسبقية أو براءة اختراع.

الملخص

تقترح هذه الورقة صياغة مركزة لثلاثية تشغيلية ضمن Moussaif AGI-Oriented Theory: HyperKB-as-OS بوصفها معرفة حاكمة قابلة للاسترجاع والتنفيذ، وHyperSkills بوصفها وحدات قدرة تعاقدية قابلة للتركيب والاختبار، وHyperLoops بوصفها حلقات عمل محدودة بالميزانية والدليل وشروط التوقف والرجوع. تنطلق الدراسة من أحدث اتجاهات Agentic AI حتى 14 يوليو 2026، حيث انتقل المجال من نمط Prompt ثابت إلى Learned Orchestration، وSkill Libraries ذات دورة حياة، و-Agent Native Memory، وتخصيص Adaptive Test-Time Compute، ومقاييس تنفيذية طويلة الأفق. ومع ذلك، تكشف Benchmarks الحديثة أن الوكلاء ما زالوا يفقدون القيود، ويكررون الفشل، ويتوقفون مبكرا أو متأخرا، ولا يثبتون أثر المهمة بصورة مستقرة.

تدافع الورقة عن أن الابتكار المساعفي ليس في وجود الذاكرة أو الأداة أو التكرار منفردا، فلكل منها سوابق واضحة، بل في تحويلها إلى عقود مترابطة داخل Kernel محايد تجاه النموذج والمزوّد: كل ذرة HyperKnowledge تحمل مصدرها وحدودها وسياساتها وروابطها التنفيذيـة؛ كل HyperSkill تحمـل Trigger وInputs وOutputs وEvidence Gate وPermission Gate وVerification وRollback؛ وكل HyperLoop تضبط التكرار بميزانية وفائدة هامشية ومحكمة تحقق وأثر عالم. ويُقترح أن هذه الثلاثية قد ترفع القابلية للتركيب والنقل بين المجالات، وتخفض استهلاك الرموز، وتحسن سلامة الذاكرة، وتجعل التطور الوكيلي قابلا للقياس والتكذيب.

تقدم الورقة تعريفا رسميا للأدوات الثلاث، ومقارنة بحالة الفن، ومعمارية دمج، ومقاييس مردود، وبرنامج تجريبي قائم على Baselines وAblations وLong-Horizon Evaluation. الخلاصة المركزية أن Moussaif Theory تستفيد من الثلاثية بقدر ما تتحول من قاموس نظري واسع إلى نظام تشغيل معرفي يختار أقل معرفة لازمة، وأصغر مجموعة مهارات كافية، وأقصر حلقة موثقة قادرة على إحداث World Effect قابل للاستعادة.

الكلمات المفتاحية: Agentic AGI؛ Moussaif Theory؛ HyperSkills؛ HyperKnowledge؛ KB-as-OS؛ HyperLoops؛ Skill Library؛ Agent Memory؛ Learned Orchestration؛ Adaptive Compute؛ Verification؛ World Effect؛ Sparse HyperRouting.

English Technical Abstract

This paper formalizes three constructs in the Moussaif AGI-Oriented Theory: HyperKB-as-OS as governed executable knowledge, HyperSkills as composable capability contracts, and HyperLoops as budgeted, evidence-gated execution cycles. It relates them to the July 2026 state of the art in learned orchestration, skill evolution, agent-native memory, adaptive test-time compute, and long-horizon benchmarks. The proposed novelty is their integration into a model-neutral kernel: knowledge atoms carry provenance and policy; skills carry verification, permissions, and rollback; loops carry stopping, budget, safety, and world-effect conditions. The paper defines the architecture, falsifiable hypotheses, ablations, and a deployment roadmap. AGI-oriented is a research target for transfer and governed agency, not a claim of achieved AGI.

خريطة الورقة

1. مقدمة وسؤال البحث

2. الحالة المعرفية ومنهج مقارنة حالة الفن

3. حالة الفن: Evaluation، Protocols، Loops، Memory، Skills، Orchestration

4. التعريف الرسمي لـHyperSkills

5. التعريف الرسمي لـHyperKB-as-OS

6. التعريف الرسمي لـHyperLoops

7. معمارية التكامل داخل Moussaif AGI-Oriented Kernel

8. موضع الأصالة مقارنة بالأعمال السابقة

9. الصياغة الرياضية ومقاييس المردود

10. الفوائد المباشرة لـMoussaif Theory

11. برنامج الاختبار والـAblation

12. محكمة الهدم والفشل المتوقع

13. خارطة التنفيذ والنشر

14. الخاتمة والمراجع

المبدأ الحاكم: لا تُحسب الوحدة ذكاءً لأنها موجودة؛ تُحسب فقط إذا أضافت فرقا موثقا في النتيجة أو الكلفة أو السلامة أو قابلية النقل.

1. مقدمة وسؤال البحث

أصبح من الممكن اليوم بناء وكلاء يخططون، ويستدعون أدوات، ويقرؤون مستودعات، ويعدّلون ملفات، ويستفيدون من ذاكرة خارجية. لكن جمع هذه القدرات لا يضمن نظاما عاما متماسكا. فالنظام قد يمتلك آلاف الأدوات ولا يعرف متى يمتنع عن استعمالها؛ وقد يمتلك ذاكرة ضخمة لكنها تعيد معلومات قريبة دلاليا وغير صالحة قراريا؛ وقد يكرر النقد والتحسين من دون تقدم هامشي أو شرط توقف. الفجوة ليست نقص مكوّن واحد، بل غياب نظام تشغيل يربط المعرفة والقدرة والتكرار بعقود مشتركة.

ينطلق برنامج مساعف من الفصل بين Model Intelligence وSystem Intelligence. الأول قدرة كامنة في الأوزان والسياق، والثاني قدرة تشغيلية ناشئة من السؤال الحاكم، والتوجيه، والذاكرة، والأدوات، والمحاكم، والصلاحيات، والقياس. ومن هنا يُستعمل وصف AGI-Oriented لا بوصفه إعلان وصول، بل بوصفه معيار تصميم: هل يستطيع النظام نقل الخبرة، وتركيب قدرات جديدة، والمحافظة على حالة طويلة، والتعلم من الأثر، والعمل عبر نماذج ومزودين مختلفين ضمن حوكمة قابلة للرجوع؟

1.1 سؤال البحث المركزي

هل يرفع دمج **HyperKB-as-OS**، **HyperSkills**، **Bounded HyperLoops** قدرة الوكيل على النجاح طويل

الأفق والنقل بين المجالات، مقارنة بـ**RAG** أو **ReAct** أو **Skill Library** أو **Reflection** كل منها منفردا، مع كلفة ومخاطر أقل؟

1.2 الفرضية المركزية

الفرضية هي أن القدرة الوكيلية الفعالة تنشأ من اقتران ثلاث وظائف: معرفة منتقاة تشرح ما ينبغي فعله ولماذا وتحت أي حدود، مهارة تعاقدية تحول المعرفة إلى قدرة قابلة للاستدعاء والاختبار، وحلقة محدودة تحول القدرة إلى تقدم تراكمي مع توقف وتحقيق ورجوع. إذا غابت أي وظيفة، ظهر أحد الأنماط الثلاثة: موسوعة لا تعمل، أداة بلا حاكمية، أو تكرار بلا تقارب.

2. الحالة المعرفية ومنهج مقارنة حالة الفن

تُعامل مصطلحات HyperSkill وHyperKB-as-OS وHyperLoop هنا كتسميات وصياغات أصلية مقترحة داخل corpus مساعف. لكن عناصرها الأولية لها سوابق في الأدبيات: RAG، Skill Libraries، Workflow Graphs، Tool Use، وAgent Runtimes، Verifiers، Search، Reflection، Graph Memory. لذلك لا تقوم دعوى الأصالة على القول إن أحدا لم يستعمل مهارة أو ذاكرة أو حلقة، بل على البنية العقدية الموحدة، وآلية التوجيه المتناثر، واقتران الدليل والصلاحية والرجوع والأثر في كل مستوى.

اعتمدت المقارنة ثلاثة مستويات. المستوى الأول Foundational Prior Art من 2022-2024 مثل ReAct وToolformer وReflexion وVoyager وMemGPT. المستوى الثاني 2025-2026 ويشمل Agent-Native Skill Evolution، Long-Horizon Benchmarks، Adaptive Compute، Learned Orchestration، Memory. المستوى الثالث هو corpus مساعف الداخلي الذي صاغ M-HKOS وHyperSkill-Graph وBounded HyperLoop وMEHDT-OS. كل نتيجة في هذه الورقة تُصنّف إما حقيقة منشورة، أو استنتاج مقارن، أو فرضية مساعفية تحتاج اختبارا.

ما يسمح به	المعنى	الرتبة
الشرح والبناء النظري فقط	تعريف أو تسمية مقترحة	E0
فرضية معقولة لا إثبات للنظام	تشابه أو دعم مكوّن من أدبيات سابقة	E1
ادعاء محلي ضمن البيئة	Prototype أو تجربة داخلية قابلة لإعادة التشغيل	E2
ادعاء أداء محدد	Benchmark مضبوط مع Baseline وAblation	E3
دليل قوي على العمومية الجزئية	تكرار مستقل ونقل بين بيئات	E4

3. حالة الفن حتى 14 يوليو 2026

3.1 من الوكيل الثابت إلى Learned Orchestration

مثّل ReAct نقطة تأسيسية بدمج التفكير والفعل والملاحظة في مسار واحد [7]. ثم اتجه المجال إلى Orchestrators تختار النماذج والأدوار ومسارات التحقق ديناميكيا. يقدّم Sakana Fugu في 2026 عائلة Learned LLM Orchestrators تبني Agentic Scaffolds متكيفة مع السؤال، وقد أبلغت عن نتائج قوية عبر مهام برمجية واستدلالية متعددة [1]. هذه النقلة مهمة لMoussaif Theory لأنها تدعم فكرة أن جودة النظام ليست مرهونة بنموذج واحد، بل بقدرة Meta-Controller على تشكيل فريق ومسار مناسبين للمهمة.

لكن Learned Orchestration لا تحل تلقائيا مشكلة الذاكرة الملوثة أو المهارة غير الموثقة أو الحلقة غير المحدودة. كما أن التعلم من Orchestration Traces يثير مشكلة Credit Assignment: أي قرار في التوجيه صنع الفرق؟ وأي وكيل تسبب في الخطأ؟ لذلك تحتاج الثلاثية المساعفية إلى Ledger يربط اختيار المعرفة والمهارة والحلقة بالنتيجة الفعلية.

3.2 Skill Libraries تتحول من أرشيف إلى دورة حياة

قدّم Voyager مكتبة مهارات برمجية قابلة للتركيب والاسترجاع، مع تحسين تكراري يعتمد على أخطاء التنفيذ والتغذية الراجعة [9]. وفي 2026 انتقل الاتجاه إلى إدارة دورة حياة المهارة. تقترح MUSE-Autoskill إنشاء المهارات وتخزينها وإدارتها وتقييمها وصقلها، مع Skill-Level Memory واختبارات ووظائف نقل بين الوكلاء [2]. وتقترح SAGE دمج مكتبة المهارات في Reinforcement Learning، وأبلغت على AppWorld عن تحسن في إنجاز السيناريو مع خطوات ورموز أقل [3].

تتقاطع هذه الأعمال مع HyperSkill لكنها لا تطابقها تماما. HyperSkill في الصياغة المساعفية ليست Prompt محفوظا ولا دالة فقط؛ إنها Capability Contract يتضمن محفزات الاستدعاء، وعدم الاستدعاء، ومتطلبات المعرفة، والصلاحيات، ومخرجات ملزمة، واختبارات قبول، وفشل، ورجوع، ومقاييس، وسياسة ترقية للذاكرة. هذه الزيادة العقدية هي موضع القيمة القابل للاختبار.

3.3 الذاكرة تنتقل من RAG ساكن إلى Agent-Native Memory

قدّم MemGPT استعارة نظام التشغيل لإدارة طبقات الذاكرة والسياق الافتراضي [6]. ومع توسع الوكلاء، ظهر نقد لRAG بوصفه استرجاعا ساكنا يقرأ مقاطع من corpus دون إدارة كاملة للكتابة والصيانة والتعارض. دراسة Agent-Native Memory في 2026 تفكك النظام إلى تمثيل وتخزين، واستخراج، واسترجاع وتوجيه، وصيانة، وتقارن عددا من الأنظمة والأعباء [5]. ويقترح SelfMem أن يتعلم الوكيل استراتيجية ذاكرته نفسها باستخدام أدوات وتغذية راجعة بدلا من قالب ثابت، مع نتائج معلنة على سياقات حتى مليون رمز [4].

HyperKB-as-OS يذهب خطوة مفاهيمية أبعد من Memory Store: المعرفة لا تخزن فقط كحقيقة، بل كذرة تشغيل تحتوي Claim، Mechanism، Evidence، Boundaries، Provenance، Linked Skills، Allowed Actions، Tests

وFreshness. أي أنها تجمع Data Plane وPolicy Plane وCapability Routing في طبقة واحدة. هذه الفكرة مفيدة، لكنها تزيد تعقيد الصيانة؛ لذلك تحتاج Dedupe وExpiry وConflict Resolution وPrivacy Tiers.

3.4 من **Reflection** المفتوح إلى **Bounded Adaptive Loops**

قدّمت Reflexion التعلم من التغذية الراجعة النصية دون تحديث الأوزان [8]، وظهرت بعد ذلك أشكال Tree Search و-Self Refinement وVerifier-Guided Search. في 2026 يبرز اتجاه تخصيص الحساب تكيفيا: TrACE يوزع الاستدعاءات حسب اتفاق المسارات [17]، وInference-Time Budget Control يخصم كلفة الأدوات والرموز ويوقف البحث عند نفاد الميزانية أو تحقق شرط الإنهاء [18]، وأعمال أخرى تخصص التحقق للحالات الأعلى عدم يقين بدلا من التحقق الموحد [19].

هذه الاتجاهات تدعم جوهر Bounded HyperLoop: ليست القيمة في تكرار النموذج، بل في تصميم Controller يقرر الملاحظة التالية، والفعل، والمتحقق، وميزانية الدورة، وكسبها الهامشي، ونقطة الاستعانة بالبشر، والرجوع. كما أن دراسات حديثة عن وهم التصحيح الذاتي تشير إلى أن النموذج قد يصحح كلام الآخرين أفضل من أثره الخاص، ما يبرر استقلال محكمة التحقق عن المنفذ عندما تكون الكلفة أو المخاطر مرتفعة [20].

3.5 البروتوكولات والاختبارات تكشف فجوة النظام

يسهّل Model Context Protocol فصل الأدوات والموارد والقوالب عن تطبيق الوكيل، ويوفر واجهة معيارية للاتصال بأنظمة خارجية [15]. وهذا يخدم قابلية نقل HyperSkills، لكنه لا يحدد وحده متى تُستدعى الأداة ولا ما الدليل المطلوب ولا كيف تُدار الذاكرة. أما الاختبارات الحديثة فتكشف أن التحدي انتقل إلى أفق المهمة: OSWorld 2.0 يضم تدفقات واقعية طويلة تتطلب مئات الاستدعاءات، وأفضل الأنظمة المعلنة ما تزال بعيدة عن الإتمام المهني [13]. كما أن SWE-Bench Pro وSWE-EVO يبرزان ضعف الوكلاء في تغييرات برمجية طويلة متعددة الملفات مقارنة بمهام الإصلاح الأقصر [14، 21].

الدرس لحالة الفن هو أن Agent Scaffold الجيد لم يعد مجرد Prompt + Tools. إنه Scheduler وMemory Manager وSkill Registry وVerifier وBudget Controller وObservability Ledger. وهذه بالضبط المساحة التي تسعى الثلاثية المساعفية إلى تنظيمها في طبقة واحدة قابلة للنقل.

الفجوة المتبقية	ما أثبتته الأدبيات	الممثلون	محور حالة الفن
اختيار الأداة والحوكمة طويلة الأفق	فائدة التفاعل مع الأدوات والبيئة	ReAct، Toolformer	التفكير والفعل
التحقق، التكرار، التلوث، حدود الصلاحية	إعادة استخدام الخبرة والمهارات	Reflexion، Voyager	التعلم من التجربة
عقود صلاحية وصلاحيات ونقل مضبوط	إدارة دورة حياة وتحسن في مهام محددة	MUSE-Autoskill، SAGE	مهارات متطورة

محور حالة الفن	الممثلون	ما أثبتته الأدبيات	الفجوة المتبقية
ذاكرة وكيلية	MemGPT، SelfMem، Agent-Native Memory	ذاكرة متعددة الطبقات واستراتيجيات متكيفة	سياسات القرار وربط المعرفة بالتنفيذ
Orchestration	Fugu، Multi-Agent RL	تشكيل مسارات ووكلاء حسب المهمة	Credit Assignment ودين التنسيق
Adaptive Loops	Budget Control، Selective Verification	تخصيص الحساب والتوقف التكيفي	تعريف تقارب شامل وقابل للرجوع
Evaluation	OSWorld 2.0، SWE-Bench Pro	قياس التنفيذ طويل الأفق	Benchmarks للتعلم المستمر والنقل والسلامة

4. HyperSkills: من مهارة نصية إلى Capability Contract

4.1 التعريف

HyperSkill هي أصغر وحدة قدرة قابلة للتركيب في Moussaif Kernel، شرط أن يكون لها تعريف نجاح، وحدود استعمال، ومدخلات ومخرجات، وأدوات مسموحة، ودليل مطلوب، وآلية تحقق، وفشل متوقع، ورجوع، وإصدار. كلمة Hyper لا تعني الحجم أو التعقيد، بل كثافة العقد: الوحدة تجمع المعرفة والتنفيذ والحوكمة والقياس في كيان واحد.

```
HS = <Purpose, Triggers, NonTriggers, Preconditions, Inputs, Outputs,
Procedure, Tools, EvidenceGate, PermissionGate, Verification, Stop, Failure,
Rollback, Metrics, Version>
```

4.2 الفرق عن Prompt، Tool، Workflow

الكيان	ما يملكه	ما ينقصه مقارنة بـHyperSkill
Prompt	تعليمات لغوية	لا يضمن Schema أو تحقق أو إصدار أو Rollback
Tool/API	وظيفة قابلة للاستدعاء	لا يحدد الحاكمية أو السياق أو جودة النتيجة
Workflow	تسلسل خطوات	قد يكون جامدا ولا يتعلم من الأثر
Skill Library Item	خبرة قابلة لإعادة الاستخدام	قد تفتقر إلى الصلاحيات والاختبارات وحدود النقل

الكيان	ما يملكه	ما ينقصه مقارنة بـHyperSkill
HyperSkill	عقد قدرة كامل	تحتاج Registry وصيانة وBenchmarks لمنع التضخم

4.3 دورة حياة HyperSkill

- Induction: استخراج مرشح مهارة من مهمة ناجحة أو حاجة متكررة.
- Compilation: تحويل الخبرة إلى عقد مستقل عن المثال الأصلي.
- Unit Tests: بناء حالات نجاح وحدود وفشل ومقاومة حقن.
- Registration: إدراجها في Skill Graph مع بصمة دلالية ووظيفية.
- Routing: استدعاؤها فقط عندما تتجاوز منفعتها المتوقعة كلفة التفعيل.
- Runtime Learning: إلحاق Trace مختصر بالمهارة لا بالسياق العام فقط.
- Promotion أو Deprecation: الترقية بعد انتقال مثبت، أو خفض الأولوية عند التقادم والفشل.

4.4 فائدة HyperSkill لـAGI-Oriented Generality

العمومية العملية لا تعني أن النموذج يعرف كل شيء مسبقا، بل أن النظام يستطيع تركيب قدرات محددة لحل مهمة جديدة. HyperSkills توفر مستوى وسطا بين وزن النموذج وأداة البيئة: أكثر تجريدا من كود خاص، وأكثر قابلية للاختبار من تعليمات حرة. وعندما تُربط في Skill Graph، يمكن إنشاء قدرات مركبة من وحدات أصغر مع قياس مساهمة كل وحدة.

5. HyperKB-as-OS: المعرفة بوصفها سياسة تشغيل

5.1 التعريف

HyperKB-as-OS ليست قاعدة معرفة ضخمة تُحقن في السياق، بل طبقة تشغيل تنظّم ما يعرفه النظام، وكيف يثق به، ومتى يستدعيه، وما الأفعال التي يسمح بها، وما المهارات والاختبارات المرتبطة به. تعمل مثل Knowledge Kernel: تحفظ الدستور، والأنطولوجيا، والذرات، والعلاقات، وسجل المصدر، وسياسة الحداثة، وخرائط المشاريع، وواجهات الاسترجاع.

```
K_atom = <Claim, Definition, Mechanism, EvidenceGrade, Sources, Freshness,
Boundaries, Counterexamples, LinkedSkills, AllowedActions, RiskClass, Tests,
Provenance>
```

5.2 لماذا OS وليس RAG فقط؟

RAG يجيب غالبا عن سؤال: ما المقاطع الأقرب إلى الاستعلام؟ أما HyperKB-as-OS فتضيف: ما السؤال الحاكم؟ ما رتبة الدليل؟ هل المعلومة صالحة لهذا المشروع؟ ما الصلاحيات؟ أي Skill يمكنها تنفيذها؟ ما الاختبار الذي يثبت الأثر؟ وهل يجب تحديث الذرة أو تجميدها؟ لذلك تُفهم OS هنا كاستعارة معمارية لإدارة الحالة والسياسة والموارد، لا كنظام تشغيل عتاد.

طبقة	وظيفتها	آلية الحماية
Constitution	الهوية والحدود والأولويات	Authority Tokens وReversibility
Ontology	تعريف الكيانات والعلاقات	Conflict Rules وVersioning
Evidence Store	الدعاوى والمصادر والحداثة	Evidence Grades وProvenance
Project Context	خرائط الأهداف والأصول والقرارات	Privacy Tiers وIsolation
Skill Links	ربط المعرفة بالقدرات	Permission Gates وCompatibility
Retrieval Router	اختيار أقل ذرات كافية	Diversity وSparse Top-K
Maintenance	دمج، نسيان، تقادم، ترقية	Dedupe، Expiry، Regression Tests

5.3 مشكلة التضخم المعرفي

أخطر فشل هو تحويل HyperKB إلى corpus ضخم مكرر. الزيادة في الصفحات أو الأسطر لا تساوي زيادة في القدرة. يجب أن تقاس كل ذرة بالفرق الوظيفي: هل تغير قرارا؟ هل تمنع خطأ؟ هل تضيف آلية أو أداة أو اختبارا؟ إذا لم تفعل، تدمج أو تؤرشف. لذلك يقترح النظام Novelty Ledger وSemantic Dedupe وUsage-Weighted Retention وFreshness Decay.

5.4 فائدة HyperKB لـ AGI-Oriented Continuity

تمنح HyperKB النظام ذاكرة مؤسسية عابرة للنماذج والجلسات. يمكن استبدال النموذج مع بقاء العقود والقرارات والمهارات. كما تسمح بضغط السياق: بدلا من تحميل المشروع كله، يسترجع الموجّه كبسولة تحتوي القيود والدليل والمهارات المرتبطة. هذا يرفع Cross-Model AGI Density لأن الذكاء التشغيلي لا يبقى محبوسا في وزن مزود واحد.

6. **HyperLoops**: التكرار المحكوم بوصفه وحدة تعلم وتنفيذ

6.1 التعريف

HyperLoop هي دورة حالة تتكون من Observe - Diagnose - Decide - Act - Verify - Update، لكنها لا تُسمح بالاستمرار إلا داخل ميزانية محددة وبشرط تقدم هامشي وسلامة وصلاحية. تختلف عن Loop عادي في أنها تحمل عقدا صريحا للهدف والحالة والمقاييس ونقطة الرجوع ومخرج التعلم. وهي Bounded افتراضيا؛ كل حلقة بلا حد أقصى أو Stop Condition تعتبر عيبا تصميميا.

```
HL = <Objective, State, Observe, Decide, Act, Verify, MaxIterations,
TokenBudget, TimeBudget, GainFloor, ApprovalPoints, StopWhen, Rollback,
LearningOutput>
```

6.2 شروط التوقف

- Goal Closure: تحقق اختبار القبول النهائي.
- Budget Exhaustion: نفاد الرموز أو الزمن أو المال.
- Marginal Gain Floor: انخفاض التحسن تحت حد محدد لدورتين أو أكثر.
- Convergence: استقرار الحالة وعدم ظهور فرق جديد.
- Safety or Authority Boundary: الحاجة إلى إذن أو ظهور خطر.
- Contradiction Escalation: تعارض أدلة لا تستطيع الحلقة حسمه.
- Human Handoff: انتقال واع إلى المستخدم مع الحالة والخطوة التالية.

6.3 استقلال التحقق

لا ينبغي أن يكون المنفذ هو المتحقق الوحيد في المسارات الحساسة. يمكن استخدام نموذج آخر، أو Test Harness، أو Rule Engine، أو قراءة حالة النظام، أو مقارنة ملفية. قانون الأثر المساعفي هو: لا يُعد الفعل منجزا لأن الوكيل قال إنه أنجزه؛ يلزم State Readback أو Test Result أو External Receipt. هذه القاعدة تحول HyperLoop من محادثة انعكاسية إلى حلقة سيبرانية مغلقة.

6.4 فائدة **HyperLoop** لـ**AGI-Oriented Adaptation**

التعلم أثناء المهمة يحتاج تكرارا، لكن التكرار غير المحكوم يضاعف الخطأ والكلفة. HyperLoop تمنح النظام قدرة على التصحيح والتجريب والاستكشاف مع حدود واضحة. وعندما تنتج الحلقة Learning Output منظما، يمكن ترشيحه إلى HyperKB أو HyperSkill بعد Regression Gate، فتتحول الخبرة إلى قدرة قابلة لإعادة الاستخدام بدلا من بقاءها أثرا طويلا في سجل محادثة.

Moussaif AGI-Oriented Kernel معمارية التكامل داخل .7

Task → Q-MCD → Sparse HyperKB Retrieval → HyperSkill Binding → Bounded HyperLoops → Independent Court → World Effect → Memory Distillation

7.1 مسار التنفيذ المرجعي

1. يحوّل Goal-Contract Compiler طلب المستخدم إلى هدف، قيود، صلاحيات، واختبارات قبول.
2. يولد Q-MCD سؤالا حاكما يقلل الغموض ويحدد نوع الفرق المطلوب.
3. يسترجع HyperKB أقل مجموعة ذرات تغطي التعريف والآلية والدليل والحدود.
4. يحسب Sparse HyperRouter ملاءمة المهارات ويختار أصغر Skill Set كاف.
5. يجمّع Loop Compiler حلقة أو أكثر بميزانيات ومتحققات مستقلة.
6. ينفذ HyperAgents الأدوار دون أن يفقد Main Brain سلسلة المصدر.
7. تفصل Falsification Court بين جودة الصياغة وصحة الأثر.
8. يقيس World-Effect Meter تغير الحالة الخارجية.
9. تقطّر Self-Reconstruction Memory الدرس إلى ذرة أو مهارة أو سياسة، أو تسجله كفشل غير قابل للتعميم.

7.2 طبقات النظام

المستوى	المكوّن	السؤال الذي يحكمه
L0	Constitution & Authority	ما المسموح وما الأولوية؟
L1	HyperKB-as-OS	ما المعرفة اللازمة وما رتبتها؟
L2	HyperSkill Graph	ما القدرة الأقل الكافية؟
L3	HyperLoop Controller	كيف نتقدم ونتوقف ونتراجع؟
L4	HyperAgents / Tools	من ينفذ وبأي أداة؟
L5	Courts & Verifiers	كيف نعرف أن النتيجة صحيحة؟
L6	World Effect & Memory	ماذا تغير وما الذي يستحق التعلم؟

7.3 الخوارزمية المختصرة

```
def moussaif_run(task, context, budget):
    contract = compile_goal_contract(task, context)
    q = qmcd(contract)
    atoms = hyperkb.retrieve(q, top_k=budget.kb_k, diverse=True)
    skills = router.bind_minimal_skills(q, atoms, budget)
    loops = loop_compiler.build(skills, contract, budget)
    traces = execute_with_permissions(loops)
    verdict = independent_court.verify(traces, contract.acceptance_tests)
    effect = world_state.readback(verdict)
    memory.promote(effect, only_if_reusable_and_regression_safe=True)
    return effect
```

8. موضع الأصالة مقارنة بالأعمال السابقة

تُفهم الأصالة هنا كتركيب نظري وهندسي يحتاج تحقق أسبقية متخصصا إذا أريد استعماله قانونيا. المقارنة التالية لا تدعي غياب كل تشابه عالمي، بل تحدد الفرق المقترح الذي يمكن إخضاعه للتجربة.

المفهوم المساعفي	السابق الأقرب	الامتداد المقترح	اختبار الأصالة الوظيفية
HyperSkill	Voyager Skills، MUSE-Autoskill، Tool APIs	عقد قدرة يجمع الدليل والصلاحية والتحقق والرجوع والنسخة	هل العقد الكامل يحسن النقل ويخفض الفشل مقابل Skill عادية؟
HyperKB-as-OS	RAG، Graph Memory، MemGPT	ذرة معرفة تربط السياسة والمهارة والاختبار والأثر	هل يقل فقد القيود والتعارض مقارنة بذاكرة استرجاعية؟
Bounded HyperLoop	ReAct، Reflexion، Search، Adaptive Compute	حلقة ذات ميزانية وكسب هامشي ومتحقق مستقل وRollback	هل تقل الكلفة والانحراف مع ثبات أو تحسن النجاح؟
الثلاثية المتكاملة	Agent Frameworks وOrchestrators	سلسلة موحدة من المعرفة إلى القدرة إلى الحلقة إلى الأثر	هل يوجد Synergy يتجاوز جمع المكونات منفردة؟
Sparse HyperRouting	MoE وModel Routing وOrchestration	توجيه مشترك للمعرفة والمهارات والحلقات حسب الفرق	هل يحقق قيمة أعلى لكل رمز ومخاطرة؟

8.1 المساهمة النظرية الأهم

المساهمة الأهم ليست تكثير أسماء الوحدات، بل تغيير وحدة تصميم الوكيل. الوحدة لم تعد Prompt أو Agent منفردا، بل Contracted Capability-in-a-Bounded-Evidence Loop grounded in Governed Knowledge. هذا التحول يجعل كل ادعاء قابلا للتفكيك: يمكن إزالة HyperKB policy أو Skill Verification أو Loop Bounds وقياس الفرق. إذا لم يظهر فرق، تسقط دعوى القيمة.

9. الصياغة الرياضية ومقاييس المردود

9.1 درجة تفعيل **HyperSkill**

$$A(s|q) = R(q,s) + E(s) + H(s) + T(s) + C(s) - Cost(s) - Risk(s) - Redundancy(s)$$

حيث R ملاءمة السؤال، وE رتبة الدليل، وH النجاح التاريخي المعاير، وT قابلية النقل، وC توافق العقود. لا تُفعّل المهارة إذا لم تتجاوز أفضل Baseline بسيط أو إذا تكررت وظيفيا مع مهارة أعلى جودة.

9.2 جودة استرجاع **HyperKB**

$$KQ = Coverage \times Provenance \times Freshness \times DecisionRelevance \times Diversity / (Noise + Conflict + ContextCost)$$

الاسترجاع الجيد ليس الأعلى تشابها فقط، يجب أن يغطي أبعاد المهمة، ويحافظ على المصدر والحداثة، ويؤثر في القرار، ويتجنب استرجاع نسخ مكررة. يضاف Conflict Penalty عندما تحمل الذرات ادعاءات متعارضة بلا سياسة حسم.

9.3 كسب الحلقة واستقرارها

$$g_t = (V_{t+1} - V_t) / (tokens_t + latency_t + money_t + risk_t)$$

$$\text{Stop if: success OR budget_exhausted OR } g_t < \text{epsilon for m rounds OR safety_boundary OR unresolved_conflict}$$

تستقر الحلقة عمليا عندما يخفض الخطأ، ولا يتكرر النقد نفسه، ويقترب الاختبار من القبول. لا يفترض النموذج الرياضي استقرارا عاما، بل يقدم مؤشرات تشغيلية تمنع التكرار اللانهائي.

9.4 مردود الذكاء الوكيلي

$$Y_{AGI} = \Delta VerifiedWorldValue \times Transfer \times Reliability \times Reversibility / (Tokens + Latency + Money + ExpectedRisk + CoordinationDebt)$$

هذا المقياس يقاوم وهم أن الاستدلال الأطول أو عدد الوكلاء الأكبر يعني ذكاء أعلى. يزداد المردود فقط عندما ترتفع قيمة صحيحة متحققة في العالم، وتنتقل القدرة، وتثبت الموثوقية، ويبقى الرجوع ممكنا، مع احتساب دين التنسيق.

9.5 أثر التكامل الثلاثي

$$G_{eff} = KQ \times SkillValidity \times LoopClosure \times CourtIndependence \times WorldEffect / MemoryPollution$$

الصيغة ليست قانونا مثبتا، إنها نموذج فرضي لاختبار التفاعل. يتوقع أن يكون الناتج مضاعفا لأن ضعف أي عامل حرج قد يخفض النظام كله: معرفة خاطئة تُفسد مهارة صحيحة، ومهارة ضعيفة تُفشل حلقة جيدة، وحلقة بلا تحقق تسجل فشلا على أنه تعلم.

10. بماذا تفيد الأدوات الثلاث Moussaif AGI-Oriented Theory؟

آلية تحققها	الفائدة
تصبح MCD وQ-MCD وCourts وسياسات الذاكرة مكونات قابلة للاستدعاء بدلا من مفاهيم وصفية.	تحويل النظرية إلى Runtime
تُبنى القدرة الجديدة من Skill Graph وHyperKB، فلا يحتاج كل مجال إلى Agent كامل مستقل.	العمومية بالتركيب
تبقى المعرفة والمهارات والعقود خارج الأوزان، ويمكن تشغيلها فوق مزودين ونماذج مختلفة.	النقل بين النماذج
Sparse Retrieval وMinimal Skill Binding يمنعان حقن corpus كامل أو تشغيل فرق كبيرة بلا حاجة.	اقتصاد الرموز
لا تُكتب الخبرة إلى الذاكرة إلا بعد تحقق وفائدة وإزالة التكرار واختبار Regression.	التعلم المستمر المحكوم
HyperLoops تحفظ الحالة والقيود ونقاط التوقف وتقاوم التكرار والضياع.	التماسك طويل الأفق
Permission Gates وRollback وHuman Approval تمنع تحويل الاستدلال إلى فعل غير مصرح.	السلامة والسلطة
كل وحدة لها Baseline وAblation وMetric، يمكن حذفها إذا لم تضف فرقا.	قابلية التكذيب
تقاس كلفة الوكلاء والمحاكم والاتصالات، فلا يتحول Multi-Agent إلى تضخم شكلي.	الوعي بدين التنسيق
ينتقل معيار النجاح من جودة النص إلى تغير الحالة الفعلية واختبارها.	World-Effect Orientation

10.1 العلاقة بـ Q-MCD، MCD

يعمل Q-MCD قبل الاسترجاع ليحدد السؤال الذي يضاعف القيمة المعرفية، ثم يستخدم MCD لقياس الفرق المتوقع بين المرشحات. HyperKB توفر مادة الفرق، HyperSkills تجسد الفرق كقدرة، وHyperLoops تختبره في مسار زمني. وبذلك تصبح المعادلات الحاكمية أدوات Routing لا مجرد مقاييس نصية.

10.2 العلاقة بـ World-Effect، HAKIM-MCD

تملأ الثلاثية قلب HAKIM-MCD: Difference Extraction يستدعي ذرات المعرفة، Vascular Routing يفعّل المهارات، Candidate Paths تُدار كحلقات، Falsification Court تختبر، World-Effect يقيس، Self-Reconstruction Memory تكتب الدرس. أي أن الأدوات الثلاث تشكل طبقة التنفيذ التي تربط النظرية بالبيئة.

10.3 العلاقة بـ MICAGFI، AGI Fattah

يفهم Fattah Gap-Opening الفجوة كمتغير مفقود يمنع الآلية أو القياس أو التنفيذ. HyperKB تسجل الفجوة، HyperSkill تبني قدرة لسدها، وHyperLoop تختبر الإغلاق. أما MICAGFI فتستفيد من حياد الأدوات تجاه النموذج: يمكن توجيه Reasoning Models أو Open-Weight Models أو خدمات متخصصة، مع حفظ النواة والعقود والذاكرة.

11. برنامج الاختبار والـ Ablation

11.1 خطوط الأساس

- B0: نموذج واحد، Prompt واحد، بلا ذاكرة أو أدوات.
- B1: RAG + Tool Calling مع سياق ثابت.
- B2: ReAct مع أدوات وذاكرة قصيرة.
- B3: Reflexion أو Self-Refine بعد الفشل.
- B4: Skill Library بلا HyperKB Policy ولا Loop Bounds.
- B5: Orchestrated Multi-Agent بلا عقود موحدة.
- M: النظام الكامل HyperKB-as-OS + HyperSkills + Bounded HyperLoops + Courts.

11.2 مصفوفة الـ Ablation

السؤال	المحذوف	التجربة
هل ترتفع الأخطاء أو المعلومات المتقادمة؟	Provenance/Freshness من HyperKB	A1

التجربة	المحذوف	السؤال
A2	NonTriggers وPermission Gate من HyperSkill	هل يزداد الاستدعاء الخاطئ والمخاطر؟
A3	Loop Budget وGain Floor	هل تتضخم الكلفة أو التكرار؟
A4	Independent Court	هل يمر التصحيح الذاتي الوهمي؟
A5	Rollback	هل يزداد الضرر عند فشل التنفيذ؟
A6	Memory Promotion Gate	هل تتلوث الذاكرة بخبرات غير قابلة للنقل؟
A7	Sparse Routing	هل تنخفض الكلفة مع الحفاظ على النجاح؟
A8	World-Effect Readback	هل تختلف جودة النص عن الإنجاز الحقيقي؟

11.3 مجالات الاختبار

يجب ألا تقتصر التجارب على أسئلة معرفة. تقترح الورقة خمس عائلات: Deep Research متعدد المصادر، Software Engineering طويل متعدد الملفات، Computer Use وتدفقات Office/Browser، Data Analysis مع ملفات واختبارات، Project Operations حيث تتغير القيود أثناء المهمة. يمكن الاستفادة من OSWorld 2.0 وSWE-Bench Pro وSWE Atlas وAppWorld، مع حزمة Moussaif Long-Horizon Suite داخل بيئة معزولة.

11.4 المقاييس

المقاييس	الفئة
Task Completion، Partial Completion، Acceptance Tests	النجاح
Tokens، Tool Calls، Latency، Monetary Cost	الكفاءة
Reproducibility، Calibration، Error Recovery	الموثوقية
Useful Recall، Pollution Rate، Conflict Rate، Dedupe	الذاكرة
Reuse، Transfer، Unit-Test Pass، Wrong Invocation	المهارات
Iterations، Marginal Gain، Premature Stop، Timeout	الحلقات

المقاييس	الفئة
Permission Violations، Rollback Success، Human Escalation	السلامة
State Readback Accuracy، User Utility، Regression-Free Change	الأثر

11.5 معايير قبول أولية

تُقبل قيمة الثلاثية فقط إذا حققت، عبر مهام جديدة لا تدريبية، تحسنا معنويا في النجاح أو خفضا واضحا في الكلفة أو المخاطر. مثال Acceptance Envelope: زيادة Task Success لا تقل عن 10% أو خفض Cost/Token لا يقل عن 20% مع عدم انخفاض النجاح؛ خفض Wrong Tool/Skill Invocation؛ رفع Rollback Success؛ والحفاظ على Transfer بعد تبديل النموذج أو المجال. الأرقام أهداف تصميمية وليست نتائج منجزة.

12. محكمة الهدم والفشل المتوقع

12.1 الفشل المفاهيمي

- تضخم التسميات: إعادة تسمية Prompt أو Workflow دون فرق وظيفي.
- ادعاء AGI من عدد الملفات أو المهارات أو الحلقات.
- الخلط بين مساحة إمكان نظرية وقدرة فعلية موثقة.
- استخدام كلمة OS مع غياب إدارة حالة أو صلاحيات أو جدولة أو صيانة.

12.2 الفشل الهندسي

- Skill Explosion: آلاف المهارات المتقاربة تضعف Routing.
- Memory Pollution: كتابة نجاحات ظاهرية أو معلومات متقادمة.
- Loop Oscillation: التبديل بين حلول بلا تقارب.
- Verifier Capture: المتحقق يكرر تحيز المنفذ نفسه.
- Coordination Debt: كلفة الوكلاء أعلى من قيمة التنويع.
- Policy Conflict: ذرة معرفة تسمح بفعل بينما الدستور يمنعه.
- Context Fragmentation: فقد العلاقة السببية أثناء الضغط والاسترجاع.
- Reward Hacking: تحسين Metric داخلي دون World Effect صحيح.

12.3 شروط رفض النظرية

يجب رفض أو تقليص الدعوى إذا لم يتفوق النظام الكامل على Baselines بعد ضبط الميزانية والنموذج والأدوات، أو إذا كان التحسن ناتجا عن سياق أطول فقط، أو إذا فشل النقل بين مهمتين، أو إذا زادت الذاكرة والمهارات دون ارتفاع التغطية الوظيفية، أو إذا لم تتنبأ درجات التوجيه بالنجاح الفعلي، أو إذا جعلت المحاكم النظام أبطأ بلا تحسن في الموثوقية.

> محكمة الهدم الحقيقية لا تسأل: هل تبدو النظرية عميقة؟ بل: ما المكوّن الذي يمكن حذفه دون أن تتغير النتيجة؟

13. خارطة التنفيذ والنشر

Minimum Research Kernel 13.1

المكوّن	أدنى تنفيذ
HyperKB	100-300 ذرة موثقة في مجالين، Graph وروابط Skill
HyperSkills	20-40 مهارة بعقود واختبارات وإصدارات
HyperLoops	6-10 قوالب حلقات: بحث، كود، تحقق، إصلاح، نشر، استرجاع
Router	Top-K sparse scoring مع سجل التوقع والنتيجة
Courts	متحقق اختباري + متحقق مصدري + متحقق صلاحيات
Memory	Promotion Queue وDedupe وRollback وExpiry
Observability	JSONL decision ledger + cost/latency/trace summaries
Benchmarks	Golden + 50 Fresh + 20 Transfer + 20 Red-Team مهمة 50

13.2 بنية الملفات المقترحة

```
moussaif-kernel/
  constitution/
  hyperkb/atoms/
  hyperkb/graphs/
  skills/HS-*.yaml
  loops/HL-*.yaml
  router/policy.yaml
  courts/
  memory/promotion.yaml
  adapters/mcp/
```

```
evals/
observability/ledger.jsonl
```

13.3 مراحل البحث

10. المرحلة 1 - Formalization: تثبيت العقود والمقاييس ومنع التكرار.
11. المرحلة 2 - Prototype: تشغيل مجال واحد مع Tests وReadback.
12. المرحلة 3 - Ablation: إزالة كل مكوّن وقياس الفرق.
13. المرحلة 4 - Cross-Model Transfer: تبديل النموذج مع بقاء النواة.
14. المرحلة 5 - Long-Horizon: مهام 100-500 خطوة مع مقاطعات وتغير قيود.
15. المرحلة 6 - Governed Evolution: إنشاء مهارات وذرات جديدة عبر Promotion Gate.
16. المرحلة 7 - External Replication: نشر Harness ومواصفات تسمح بإعادة التجربة.

13.4 الأولوية البحثية

الأولوية ليست إنشاء آلاف الملفات، بل بناء 20 مهارة و8 حلقات و200 ذرة ذات جودة عالية، ثم إثبات أن Routing يعرف متى يستخدمها وأن Memory لا تتلوث. بعد ذلك يمكن التوسع. الترتيب الصحيح هو: Contract Quality ثم Evaluation ثم Scale؛ لا Scale ثم البحث عن معنى.

14. الخاتمة

تكشف حالة الفن في 2026 أن بنية الوكيل أصبحت هي ساحة التقدم الأساسية بقدر النموذج نفسه. Learned Orchestration يختار مسارات متكيفة، Skill Libraries تتطور وتتعلم، Agent Memory يصبح قابلا للكتابة والصيانة، Adaptive Compute يخصص التكرار والتحقق، وBenchmarks طويلة الأفق تفضح فقد القيود وضعف الإنهاء. ضمن هذا السياق، تقدم ثلاثية HyperKB-as-OS وHyperSkills وBounded HyperLoops صياغة مساعفية منطقية وقابلة للاختبار.

HyperKB-as-OS تمنح المعرفة بنية حاكمية وتنفيذية، HyperSkills تحولها إلى قدرات تعاقدية قابلة للتركيب والنقل، HyperLoops تحول القدرات إلى تقدم تراكمي محدود ومتحقق. تكمن فائدة الثلاثية لـMoussaif AGI-Oriented Theory في أنها تجعل MCD وQ-MCD وHAKIM-MCD وCourts وWorld-Effect وSelf-Reconstruction Memory تعمل داخل Runtime واحد، وتحول البرنامج النظري إلى فرضيات هندسية يمكن هدمها أو تحسينها.

لا تثبت الورقة تحقق AGI، ولا أن المصطلحات وحدها اختراع مكتمل. القيمة الحقيقية ستظهر فقط إذا أثبت النظام الكامل، تحت ميزانية متساوية، نجاحا أعلى أو كلفة أقل أو نقلا أفضل أو سلامة أقوى من البدائل. ولذلك فإن العمل التالي ليس مضاعفة عدد HyperSkills أو HyperLoops، بل بناء Benchmark Harness علني، وتنفيذ Kernel صغير، وإجراء Ablations، ثم ترقية الوحدات التي تصنع فرقا موثقا.

> الزبدة: **HyperKB** يحدد ما ينبغي معرفته، **HyperSkill** يحدد ما ينبغي القدرة عليه، **HyperLoop** يثبت أن القدرة

أغلقت المهمة دون أن تفقد الدليل أو الحدود.

المراجع العلمية والتقنية

[1] Tang, Y. et al. Sakana Fugu Technical Report. arXiv:2606.21228, 2026.
[2] Lin, H. et al. MUSE-Autoskill: Self-Evolving Agents via Skill Creation, Memory, Management, and Evaluation. arXiv:2605.27366, 2026.
[3] Wang, J. et al. Reinforcement Learning for Self-Improving Agent with Skill Library (SAGE). arXiv:2512.17102, 2025/2026 revision.
[4] Yang, S. et al. SelfMem: Self-Optimizing Memory for AI Agents. arXiv:2607.03726, 2026.
[5] Are We Ready For An Agent-Native Memory System? arXiv:2606.24775, 2026.
[6] Packer, C. et al. MemGPT: Towards LLMs as Operating Systems. arXiv:2310.08560, 2023.
[7] Yao, S. et al. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629, 2022/ICLR 2023.
[8] Shinn, N. et al. Reflexion: Language Agents with Verbal Reinforcement Learning. arXiv:2303.11366, 2023.
[9] Wang, G. et al. Voyager: An Open-Ended Embodied Agent with Large Language Models. arXiv:2305.16291, 2023.
[10] Schick, T. et al. Toolformer: Language Models Can Teach Themselves to Use Tools. arXiv:2302.04761, 2023.
[11] Shen, Z. et al. GeAR: Graph-enhanced Agent for Retrieval-augmented Generation. arXiv:2412.18431, 2024.
[12] Zhang, C. et al. Reinforcement Learning for LLM-based Multi-Agent Systems through Orchestration Traces. arXiv:2605.02801, 2026.
[13] Yuan, M. et al. OSWorld 2.0: Benchmarking Computer Use Agents on Long-Horizon Real-World Tasks. arXiv:2606.29537, 2026.
[14] Deng, X. et al. SWE-Bench Pro: Can AI Agents Solve Long-Horizon Software Engineering Tasks? arXiv:2509.16941, 2025.
[15] Model Context Protocol. Specification 2025-11-25 and draft documentation, accessed 14 July 2026.
[16] Xie, T. et al. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. arXiv:2404.07972, 2024.
[17] Do not Overthink It: Inter-Rollout Action Agreement as a Free Signal for Adaptive Agent Compute (TrACE). arXiv:2604.08369, 2026.
[18] Inference-Time Budget Control for LLM Search Agents. arXiv:2605.05701, 2026.
[19] Adaptive Test-Time Compute Allocation via Learned Heuristics over Categorical Structure. arXiv:2602.03975, 2026.
[20] The Self-Correction Illusion: LLMs Correct Others but Not Themselves. arXiv:2606.05976, 2026.
[21] SWE-EVO: Benchmarking Coding Agents in Long-Horizon Software Evolution. arXiv:2512.18470, 2025/2026 revision.
[22] Establishing Best Practices for Building Rigorous Agentic Benchmarks. arXiv:2507.02825, 2025.
[23] Moussaif, Z. Moussaif1B Agentic AGI - 20 Theories Research Paper. Internal research corpus, 13 July 2026.
[24] Moussaif, Z. MEHDT-OS: HyperKB-to-HyperSkill-to-HyperLoop Governed Exponential Deployment. Internal research corpus, 13 July 2026.
[25] Moussaif, Z. Hermes-Fugu Fusion HyperSkills KB. Internal technical knowledge base, 26 June 2026.

بيان النسبة والحدود

هذه الورقة تؤسس برنامجا قابلا للاختبار، ولا تستبدل الاختبار بكثرة المصطلحات أو الملفات.

إعداد وتركيب: زهير مساعف / Zouhair Moussaif. تُنسب المصطلحات والصياغة التكاملية في هذه الورقة إلى برنامج Moussaif AGI-Oriented Theory. لا تمثل الورقة مراجعة قانونية شاملة للأسبقية أو براءة اختراع، ولا تدعي أن النظام قد اجتاز Benchmarks خارجية. جميع النتائج الرقمية المنسوبة للأعمال الأخرى هي نتائج معلنة في مصادرها، وبعض مصادر 2026 هي Preprints تحتاج مراجعة مستقلة.

الخلاصة التنفيذية الأخيرة

أقوى صيغة بحثية للأدوات الثلاث ليست: قاعدة معرفة أكبر، أو مهارات أكثر، أو حلقات أطول. بل: HyperKB أقل ضجيجا وأعلى مصدرية، HyperSkills أقل عددا وأوضح عقودا، HyperLoops أقصر وأكثر تحققًا وقابلية للرجوع.

الخطوة الحاسمة التالية هي Minimum Research Kernel صغير مع Benchmarks وAblations وDecision Ledger. عندها فقط يمكن تحديد ما إذا كانت الثلاثية تحقق زيادة حقيقية في Task Success وTransfer وIntelligence Yield.

Zouhair Moussaif - 14 July 2026 / زهير مساعف