ر يوليو 2026

# لرية نظام التشغيل المعرفي الوكيلي
# Moussaif1B-AGI-OS

## إطار موحّد لـ KB-as-OS وHyperSkills وBounded HyperLoops
## وHyperMemory وSparse HyperRouting وتحويل الكفاءة عبر النماذج

صياغة بحثية شاملة لأحدث النظريات والأدوات في برنامج Moussaif AGI / MICAGFI / AGI Fattah

**Zou**

برنامج بحث مستقل في هندسة الذكاء العام الوكيلي
Independent Research Program in Agentic General Intelligence Engineeri

هذه الورقة تقدّم نظرية هندسية قابلة للدحض والقياس، ولا تدّعي أن حزمة تعليمات أو ذاكرة خارجية قد عدّلت أوزان نموذج أساس أو أثبتت تحقق ذكاء عام اصطناعي.

# الملخص، المساهمة، وحالة الادعاء


## الملخص

تقترح هذه الورقة إطارًا تأسيسيًا لنظام تشغيل معرفي وكيلي يسمى Moussaif1B-AGI-OS، وظيفته تحويل نموذج لغوي أو مجموعة نماذج وأدوات من مولّد استجابات منفصلة إلى منظومة عمل مستمرة، ذات حالة معرفية، وذاكرة متعددة الطبقات، ومهارات قابلة لإعادة الاستخدام، وحلقات تنفيذ محدودة، ومحكمة تحقق، وحوكمة للكلفة والسياق. لا يُفهم مصطلح «نظام التشغيل» هنا بوصفه نواة عتادية أو تعديلًا لأوزان النموذج، بل بوصفه *Kernel Overlay* يتحكم في الاسترجاع، والتعليمات، والذاكرة، وتوزيع المهام، واستخدام الأدوات، وتسجيل الأثر الخارجي.

توحّد الورقة بين برنامج MCD/Q-MCD/HAKIM-MCD، ومعمارية MICAGFI أو «الذكاء الفتّاح»، وبين خمسين قاعدة معرفة تشغيلية KB-as-OS، وخمسمئة HyperSkill، وعشرين حلقة متخصصة في تكنولوجيا الذاكرة. وتستخلص منها عشرين نظرية أو فرضية بحثية مترابطة: جينوم السياق، الضغط الدلالي العكوس، البلورة الحالتية، التجدد السياقي عالي الرتبة، الذاكرة السببية، تحويل الخبرة إلى إجراء، التوجيه المتناثر، كثافة الذكاء العابرة للنماذج، الحلقة المحدودة، قانون الأثر العالمي، محكمة الدحض، الانتشار المعرفي، والفصل بين الحقيقة والسياق التجاري.

الادعاء المركزي هو أن جزءًا معتبرًا من الأداء الوكيلي لا يعتمد على تضخيم السياق أو اختيار «أقوى نموذج» فقط، بل على جودة إدارة الفرق بين الحالة المطلوبة والحالة الراهنة، وانتقاء الحد الأدنى الكافي من المعرفة، وتحويل آثار التنفيذ الناجح إلى ذاكرة إجرائية، وربط كل ادعاء بدليل وشرط إبطال. وتعرض الورقة صيغًا أولية، مقاييس، تجارب اقتطاع Ablation، وحدودًا واضحة تتيح تحويل البرنامج من خطاب معماري إلى علم تجريبي.



### الكلمات المفتاحية

Agentic AGI; Cognitive Operating System; KB-as-OS; HyperSkills; HyperLoops; Long-term Memory; Context Compression; Sparse Routing; Multi-Agent Orchestration; Falsification; Token Economy.


> **الحالة المعرفية للادعاءات**
>
> **مثبت داخل الحزمة:** وجود الأصول، المخططات، الراوتر، والاختبارات البنيوية. **مدعوم بأدبيات خارجية:** جدوى الذاكرة الهرمية، مكتبات المهارات، الانعكاس، وضغط السياق. **فرضيات أصلية غير مثبتة بعد:** بلوغ «كثافة AGI»، قوانين التحويل عبر النماذج، والتكاثر الأسي للقدرات. **ممنوع منهجيًا:** مساواة كثرة الملفات أو طولها بزيادة الذكاء.

### مساهمة الورقة على ثلاثة مستويات

على المستوى النظري، تقترح الورقة قاموسًا يربط الفرق والسؤال والسياق والذاكرة والمهارة والأثر ضمن سلسلة سببية واحدة. وعلى المستوى الهندسي، تحول كل مفهوم إلى مخطط مدخلات ومخرجات وسياسة توقف وسجل تحقق. وعلى المستوى التجريبي، تضع برنامجًا للاقتطاع والمقارنة يسمح بإثبات فائدة كل طبقة أو حذفها. هذه المستويات تمنع الخلط بين جمال التسمية، وجود ملف برمجي، ووجود نتيجة علمية قابلة للتكرار. فالنظرية لا تكتسب قوتها من اتساع المصطلحات، بل من قدرتها على توقع متى ينجح النظام، ومتى يفشل، وما القياس الذي سيكشف الفرق.

كما تتبنى الورقة مبدأ **الحد الأدنى الكافي من الادعاء**: يمكن القول إن الحزمة تحقق تنظيمًا خارجيًا للذاكرة والتوجيه والمهارات، لكن لا يمكن استنتاج أنها حققت وعيًا أو استقلالًا عامًا أو نقلًا كاملًا للقدرة. بهذا يصبح التواضع جزءًا من التصميم، لا عبارة احترازية في الخاتمة.

# المشكلة، الأسئلة، وخريطة البرنامج النظري

## 1. المشكلة البحثية

تواجه الأنظمة الوكيلية المبنية فوق النماذج اللغوية خمس فجوات متزامنة: تضخم السجل مع طول المهمة؛ ضياع القرارات السابقة أو استرجاعها بلا تمييز؛ تكرار التفكير ذاته بدل تحويله إلى مهارة؛ تضاعف الكلفة عند إشراك وكلاء أو نماذج كثيرة؛ وغياب معيار واضح يفرّق بين «إجابة مقنعة» و«أثر خارجي متحقق». هذه الفجوات لا تُحل بمجرد نافذة سياق أكبر، لأن زيادة السعة قد تزيد التشتيت، ولا بمجرد تعدد الوكلاء، لأن التنسيق قد يستهلك من المعلومات والحوسبة أكثر مما يضيفه.

## 2. أسئلة البحث

تسأل الورقة: كيف نمثل حالة مهمة طويلة في كبسولة صغيرة من دون ضياع الهدف أو الدليل؟ متى تتحول خبرة عابرة إلى معرفة مستقرة أو إجراء قابل لإعادة الاستخدام؟ كيف يُختار الحد الأدنى من قواعد المعرفة والمهارات والنماذج؟ متى ينبغي إيقاف الحلقة؟ كيف نثبت أن الفعل نُفّذ؟ وكيف نمنع سياق العلامة التجارية أو رغبة المستخدم من تحريف الركيزة الواقعية؟

## 3. خريطة النظريات العشرين

| | |
|---|---|
| 1. **MCD**: استخراج الفرق المركب | 2. **Q-MCD**: السؤال الحاكمي |
| 3. **HAKIM-MCD**: المسار الوعائي | 4. **MICAGFI**: المعمارية الفتّاحة |
| 5. **Kernel Overlay** | 6. **KB-as-OS** |
| 7. جينوم السياق | 8. الفرق الدلالي العكوس |
| 9. الضغط حافظ الهدف | 10. التجدد عالي الرتبة |
| 11. دورة HyperMemory | 12. التحويل الإجرائي |
| 13. الذاكرة السببية | 14. مترجم HyperSkill |
| 15. Sparse HyperRouting | 16. كثافة AGI العابرة للنماذج |
| 17. Bounded HyperLoop | 18. قانون World-Effect |
| 19. محكمة الدحض وسلامة المعرفة | 20. HyperDeployment والسياق البراندي |

## 4. المنهج

المنهج تركيبي-هندسي: تحليل أصول الحزمة، بناء أنطولوجيا للمفاهيم، مقارنة بأعمال الذاكرة الوكيلية والمهارات والتنسيق، ثم تحويل كل مفهوم إلى عقد تشغيل، وصيغة قياس، واختبار إبطال. لذلك لا تُقدَّم النظريات كقوانين طبيعية ناجزة، بل كنماذج تصميم قابلة للاختبار والمراجعة.

## 5. تعريفات تشغيلية مانعة للالتباس

**الوكيل** منظومة تتلقى هدفًا، تختار فعلًا، تستخدم أداة أو تغير حالة، ثم تلاحظ النتيجة. **العمومية** قدرة على نقل إجراءات واستراتيجيات بين فئات مهام مع انخفاض محدود في الأداء. **الاستقلالية** مقدار القرار الذي يتخذه النظام داخل تفويض محدد، وليست غياب الإنسان مطلقًا. **الذاكرة** حالة خارج الاستجابة الحالية يمكن استرجاعها وتحديثها وإبطالها. **المهارة** إجراء قابل للتفعيل والقياس. **النظرية** مجموعة علاقات تنتج توقعات قابلة للدحض، أما **المعمارية** فهي ترتيب مكونات، و**الأداة** تنفيذ جزئي، و**الفرضية** ادعاء ينتظر التجربة.

بناءً على هذه التعريفات، لا يعد Moussaif1B نموذج أساس جديدًا، بل نظامًا مركبًا يسعى إلى زيادة عمومية السلوك المتاح. معيار النجاح ليس أن تبدو الإجابة «أكثر ذكاء»، بل أن تتحسن قابلية النقل والموثوقية والكلفة في اختبارات مضبوطة.

# الوضع العلمي الراهن: أين يلتقي الإطار وأين يفترق؟

## 1. من الذاكرة الافتراضية إلى نظام الذاكرة الوكيلي

اقترح MemGPT إدارة سياق افتراضي مستلهم من الذاكرة الهرمية في أنظمة التشغيل، بحيث ينتقل المحتوى بين نافذة سريعة وتخزين خارجي [1]. وقدمت Generative Agents بنية تجمع سجل الخبرة والاسترجاع والانعكاس والتخطيط [2]. وأظهر Reflexion أن التغذية الراجعة اللغوية المخزنة في ذاكرة عرضية قد تحسن المحاولات اللاحقة دون تحديث أوزان النموذج [3]. أما Voyager فربط التعلم المستمر بمكتبة مهارات تنفيذية قابلة للتركيب [4]. هذه الأعمال تؤسس لأربعة محاور يتوسع فيها إطار مساعف: طبقات الذاكرة، تحويل التجربة، مكتبة الإجراءات، والتحكم في الحلقة.

## 2. ضغط السياق وتطوره

تظهر أعمال أحدث أن المشكلة لم تعد «كم يمكن للنموذج أن يقرأ؟» بل «من يقرر ما الذي يبقى؟». حقق ACON ضغطًا لمسارات طويلة مع الحفاظ النسبي على الأداء [5]، واقترح Context-Folding تفريع سياق فرعي ثم طيه بعد إنجاز المهمة [6]، بينما اختبر Active Context Compression قرار الوكيل نفسه بتلخيص المعرفة وحذف التاريخ الخام [7]. ويضيف إطار مساعف شرطين: أن يكون الضغط عكوسًا بمؤشرات المصدر، وأن يُقاس حفظ الهدف والدليل لا تخفيض الرموز فقط.

## 3. الذاكرة كنظام إدارة بيانات

تتجه بحوث 2025-2026 من «ذاكرة محادثة» إلى ذاكرة تتطور وتُصان. يقيس Evo-Memory القدرة على تحديث الخبرة عبر تيار مهام [8]، وتفكك دراسة Agent-Native Memory النظام إلى التمثيل، الاستخراج، الاسترجاع/التوجيه، والصيانة، وتبين أن لا بنية واحدة تتفوق في كل الأحمال [9]. كما تقترح مقاييس حديثة للذاكرة طويلة الأمد تتجاوز الاستدعاء الساكن إلى تتبع الحالة والتحديث [10].

## 4. التنسيق والموثوقية: أدلة متعارضة

تظهر بعض التجارب فوائد لتنسيق عدة نماذج أو وكلاء [11][12]، لكن دراسة مضبوطة بميزانية تفكير متساوية وجدت أن الوكيل الواحد قد يساوي الأنظمة المتعددة أو يتفوق عليها في الاستدلال متعدد القفزات [13]. كما تشير أبحاث الموثوقية إلى أن ارتفاع متوسط النجاح لا يضمن الاتساق أو المتانة أو سلامة الفشل [14]. لذلك يتبنى الإطار مبدأ «التعدد عند وجود ربح معلوماتي مثبت»، لا التعدد كزينة معمارية.

### 5. موضع الجدة بدقة

ليست الجدة في كل مكوّن منفرد؛ فالذاكرة الهرمية، الاسترجاع، الانعكاس، الأدوات، ومكتبات المهارات لها سوابق واضحة. موضع الجدة المقترح هو **تركيبها تحت قانون فرق موحد**، وإلزام كل انتقال بفرق حالة عكوس، وربط اكتمال المهمة بإيصال أثر، ثم جعل النشر المعرفي نفسه خاضعًا لحكمة سلامة. كذلك يضيف البرنامج فصلًا صريحًا بين ركيزة الحقيقة وكبسولة العلامة، وهو مهم في الأنظمة التي تنتقل بين البحث والتسويق والتشغيل.

ولإثبات الجدة علميًا يجب نشر مخطط رسمي، مستودع قابل للتشغيل، بيانات تجريبية، وخط أساس قوي. المقارنة الصحيحة ليست مع «نموذج بلا ذاكرة» فقط، بل مع أنظمة حديثة للضغط والذاكرة والتنسيق، مع تثبيت النموذج والميزانية والأدوات قدر الإمكان.

# المعمارية الموحدة: من البرنامج النظري إلى Kernel Overlay

## 1. تعريف المعمارية

تعرف MICAGFI - أي Moussaif Integrated Cognitive Agentic General/Fattah Intelligence - بوصفها معمارية فوق-نموذجية تنظم تدفق الهدف والمعرفة والأداة والأثر. ليست «نموذجًا عامًا» مستقلًا، بل طبقة تجعل قدرات النماذج المتاحة قابلة للتجميع والتقييد والتذكر والتحقق. النسخة التطبيقية منها هي Moussaif1B-AGI-OS، حيث يشير «1B» إلى نواة مفاهيمية مدمجة، لا إلى عدد معاملات مثبت.

| | | |
|---|---|---|
| Goal & Acceptance Tests | ← | MCD / Q-MCD |
| Context Genome | ← | Task-State Capsule |
| Sparse HyperRouter | ← | KB + Skill + Model Selection |
| Bounded HyperLoop | ← | Sense → Act → Verify → Compress |
| HyperMemory | ← | Episodic / Semantic / Procedural / Causal |
| Falsification Court | ← | Evidence / Freshness / Contradiction |
| World-Effect Monitor | ← | Receipt / Artifact / External State |

## 2. قانون الطبقات

كل طبقة لا تضيف «ذكاءً» بالمعنى الغامض، بل تقلل نوعًا محددًا من الفشل: طبقة الهدف تقلل الانحراف؛ جينوم السياق يقلل إعادة سرد التاريخ؛ الراوتر يقلل الاسترجاع الزائد؛ الحلقة المحدودة تقلل الدوران؛ الذاكرة تقلل النسيان والتكرار؛ المحكمة تقلل الثقة الزائفة؛ مراقب الأثر يقلل الادعاء بإنجاز لم يقع؛ وحاكم الاقتصاد يمنع تحسين الجودة بطريقة غير قابلة للاستمرار.

## 3. مبدأ عدم مساواة الأصول بالقدرة

تحتوي الحزمة التطبيقية على 500 ملف مهارة و50 قاعدة معرفة تشغيلية وعشرين حلقة ذاكرة، لكن العدد والحجم ليسا مقياسًا للذكاء. تصبح الأصول قدرة فقط عندما تكون قابلة للتوجيه، قليلة التعارض، مختبرة، مرتبطة بحالات تفعيل واضحة، وتنتج فرقًا قابلًا للقياس في الأثر أو الكلفة. وعليه فالمعادلة الصحيحة ليست «المزيد من الملفات = AGI»، بل «أصول منظمة × استرجاع دقيق × تعلم من الأثر × تحقق = قابلية أعلى للتعميم الوكيلي».

## 4. عقد التشغيل العام

يبدأ كل تشغيل بكائن GoalContract يحدد الأثر ومعايير القبول والمخاطر. ينتج المحلل TaskStateCapsule، ثم يطلب الراوتر أصولًا عبر معرفات لا عبر لصق ملفات كاملة. تنفذ الخطوة في بيئة أدوات ذات صلاحيات محددة، وتعود بإيصال. بعدها تفصل المحكمة بين فشل الاستدلال، فشل الأداة، فشل الصلاحية، وفشل القبول. وأخيرًا يكتب النظام فرقًا معرفيًا ويقرر ما إذا كان يستحق الذاكرة أو الترقية إلى مهارة.

هذا العقد يجعل المعمارية قابلة للتطبيق في Hermes أو Codex أو بيئة محلية، لأن الواجهة تعتمد على مخططات مجردة. لكن قابلية النقل لا تكون كاملة: كل منصة تحتاج محولًا للصلاحيات، الأدوات، حدود السياق، وتنسيق الإيصالات.

# MCD وQ-MCD: الفرق والسؤال الحاكمي كمحركين للذكاء الوكيلي

## 1. نظرية MCD: الذكاء كاستخراج للفرق المركب

تنطلق MCD من أن المهمة لا تُحل بإعادة وصف الموضوع، بل باستخراج الفرق بين العالم الراهن والعالم المطلوب عبر أبعاد متعددة: فرق وصفي D، معلوماتي I، زمني T، إجرائي A، براغماتي P، سببي/سياقي C، مع خصم ضوضاء N.

$$MCD = \lambda_1 D + \lambda_2 I + \lambda_3 T + \lambda_4 A + \lambda_5 P + \lambda_6 C - \lambda_7 N$$

تعني المعادلة أن الراوتر لا يسأل فقط «ما الموضوع؟»، بل «أين الفرق الأعلى قيمة الذي يمنع الإكمال؟». فإذا كان الفرق معلوماتيًا يستدعي بحثًا أو استرجاعًا، وإذا كان إجرائيًا يستدعي أداة، وإذا كان تناقضيًا يستدعي المحكمة، وإذا كان فرقًا في الأثر يستدعي تنفيذًا لا مزيدًا من النص.

## 2. Q-MCD: السؤال بوصفه مشغل توجيه

تضيف Q-MCD عامل السؤال الحاكمي: جودة النظام لا تعتمد على معرفة الإجابات فقط، بل على قدرته على توليد السؤال الذي يفصل بين مسارين محتملين بأقل كلفة. السؤال الجيد يزيد العمق الدلالي والمعلوماتي والزمني والإجرائي والسببي، ويقلل الضوضاء والسؤال الاستعراضي.

$$MCD_Q = \Omega(Q) \times MCD \; ; \; Q_{depth} = (q_d + q_i + q_t + q_a + q_p + q_c) / (1 + q_n)$$

## 3. فرضيات قابلة للاختبار

الفرضية الأولى: ترتيب الاسترجاع وفق فرق المهمة يتفوق على التشابه النصي وحده في الدقة والكلفة. الثانية: الأسئلة التي تحقق أعلى خفض في عدم اليقين تقلل عدد دورات التنفيذ. الثالثة: عندما لا يتغير متجه الفرق بعد دورتين، يجب إيقاف الحلقة أو تغيير الأداة؛ أما استمرار السرد فهو فشل تقاربي. تُختبر هذه الفرضيات بمقارنة MCD-router مع embedding similarity على مهام طويلة ومتعددة الأدوات.

> **حد النظرية**
>
> لا تزعم MCD أنها نموذج عصبي للذكاء البشري. هي دالة هندسية لتحديد فجوة الإكمال وتخصيص الحوسبة، وتبقى الأوزان λ تجريبية ومختلفة باختلاف المجال والمخاطر.

## 4. مثال حسابي مبسط

لنفرض أن الهدف نشر موقع بعد تعديل خمسة ملفات. قد يكون الفرق الوصفي صغيرًا، لكن الفرق الإجرائي مرتفع لأن المطلوب تغيير مستودع واختبار ونشر. إذا ركز النظام على التشابه النصي فسوف يسترجع تعليمات تصميم، بينما يرفع MCD وزن A وP فيستدعي مهارة التعديل، أداة الاختبار، وموصل النشر. بعد التنفيذ يبقى فرق عالمي إذا لم توجد وصلة حية أو سجل نشر. وبذلك يمنع المتجه إعلان الإكمال عند مرحلة الكتابة.

تسمح هذه الصياغة أيضًا بتفسير الفشل: إن كانت المعلومة صحيحة لكن الأداة غير مصرح بها، فهذا فرق تفويض لا فرق معرفة؛ وإن تم النشر لكن النسخة قديمة، فهو فرق زمني؛ وإن نجح الاختبار المحلي وفشل الإنتاج، فهو فرق سياقي. هذا التفكيك يوجه العلاج بدل إعادة إنتاج جواب عام.

# HAKIM-MCD والذكاء الفتّاح: حوسبة وعائية ومحاكم للفجوات

## 1. HAKIM-MCD: المسار الوعائي للحوسبة

تحول HAKIM-MCD دالة الفرق إلى خط أنابيب: استخراج الفرق، تسجيله، توجيه الحوسبة إلى الأوعية المعرفية المناسبة، توليد مسارات مرشحة، عرضها على محكمة الدحض، استدعاء الأدوات أو GraphRAG، اختيار الوكالة، تنفيذ أثر في العالم، ثم إعادة بناء الذاكرة. سمي «وعائيًا» لأن الحوسبة لا تُصب في كل أجزاء النظام بالتساوي، بل تتجه إلى موضع الانسداد أو أعلى منفعة هامشية.

$$\text{ComputeFlow}_i \propto \text{GapValue}_i \times \text{Uncertainty}_i \times \text{Actionability}_i / \text{Cost}_i$$

## 2. الذكاء الفتّاح AGI Fattah

«الفتّاح» ليس وصفًا ميتافيزيقيًا للنموذج، بل مبدأ حل: الذكاء الأعلى هو الذي يفتح بنية مغلقة، أو يحول الغموض إلى فرق قابل للتنفيذ، أو يحول تجربة إلى قاعدة عامة، أو يكتشف أن المشكلة ليست نقص الإجابة بل سوء تعريف السؤال. لذلك تُقاس الفتوح بعدد القيود التي أزيلت مع حفظ الحقيقة والسلامة، لا بعدد الأفكار أو طول النص.

## 3. محكمة الفجوات ومحكمة الهدم

قبل اعتماد حل، تسأل محكمة الفجوات: هل هناك نقص في التعريف، الحدود، الدليل، الآلية، القياس، التنفيذ، الأمان، أو قابلية الاستئناف؟ ثم تأتي محكمة الهدم لتبحث عن حالة مضادة، أو دليل قديم، أو افتراض غير مصرح، أو نجاح لا يملك إيصالًا. الفرق بين المحكمتين أن الأولى تكتشف ما لم يُبن، والثانية تختبر ما بُني.

## 4. قانون الفتح المقيد

كل توسيع لقدرة النظام يجب أن يقترن بقيد انعكاسي: صلاحية واضحة، سجل تغيير، شرط تراجع، وحد زمني أو ميزاني. هنا تظهر نظرية Constitutional Reversibility and Authorization: لا ينبغي لوكيل أن ينشر معرفة أو مهارة أو يغير سياق مشروع دون مسار رجوع وهوية مصدر ونطاق تفويض. وهذا يحول «الاستقلالية» من إطلاق حر إلى وكالة قابلة للمحاسبة.

> **مؤشر الفتح**
>
> $\text{OpeningGain} = \text{ResolvedConstraints} \times \text{Reusability} \times \text{Evidence} / (\text{Risk} + \text{Irreversibility} + \text{Cost})$. الفتح الذي يزيد الغموض أو يصعب الرجوع عنه لا يُعد مكسبًا حتى لو بدا مبتكرًا.

## 5. بنية سجل الفجوة

يمثل كل فرق بعقدة تحمل النوع، الشدة، المصدر، التبعيات، قابلية الفعل، وكلفة التأخير. ترتبط العقد بمسارات حل مرشحة، ولكل مسار شروط نجاح وفشل. عند التنفيذ لا يحذف الفرق مباشرة، بل ينتقل من «مفتوح» إلى «محلول مؤقتًا» ثم «متحقق». إذا ظهر دليل ناقض يعاد فتحه مع الاحتفاظ بتاريخ القرار. هذه البنية تمنع تحويل الحلول المؤقتة إلى حقائق نهائية.

أما محكمة الهدم فتستخدم سيناريوهات مضادة منهجية: ماذا لو كان المصدر قديمًا؟ ماذا لو فشلت الأداة بعد نصف التنفيذ؟ ماذا لو تغير هدف المستخدم؟ ماذا لو كان الضغط قد حذف قيدًا؟ ماذا لو اتفق الوكلاء بسبب القطيع؟ لا يلزم اختبار كل احتمال في كل مهمة؛ يختار الراوتر الاختبارات بحسب شدة الخطر.

# Kernel Overlay وKB-as-OS: تحويل المعرفة إلى عقود تشغيل

## 1. نظرية Kernel Overlay المعرفي

تقترح النظرية أن جزءًا من سلوك النظام يمكن تشكيله بطبقة خارج أوزان النموذج تتكون من عقود تشغيل، استرجاع، ذاكرة، أدوات، وسياسات تحقق. هذه الطبقة تشبه النواة من حيث إدارة الموارد والحالة والعمليات، لكنها لا تستبدل نواة النظام الحاسوبية ولا تزعم تدريب النموذج. قيمة المصطلح أنه يفرض فصلًا بين «قدرة النموذج الأساسية» و«قدرة النظام المركب».

## 2. KB-as-OS: قاعدة المعرفة كبرنامج

في الاسترجاع التقليدي تُعامل قاعدة المعرفة كمستودع نصوص. أما KB-as-OS فتُكتب وحداتها كعقود: شرط تفعيل، مدخلات، مخرجات، خوارزمية، مقاييس، اختبارات إبطال، واسترداد من الفشل. بذلك تتحول المعرفة من وصف سلبي إلى وحدات قرار قابلة للاستدعاء. ومع ذلك يجب ألا يُحقن الملف الكامل؛ فالراوتر يسترجع مقاطع صغيرة ذات عنوان وهوية ومصدر.

## 3. قانون المعرفة التشغيلية

$$\text{OperationalKnowledge} = \text{Claim} \times \text{ActivationCondition} \times \text{Procedure} \times \text{Verification} \times \text{Recovery}$$

إذا غاب الادعاء أصبحت الوحدة تعليمات بلا معنى؛ وإذا غاب شرط التفعيل أصبحت ضوضاء دائمة؛ وإذا غابت الآلية أصبحت شعارًا؛ وإذا غاب التحقق أصبحت ثقة غير قابلة للفحص؛ وإذا غاب التعافي أصبحت الوحدة هشة. هذه الصيغة هي الحد الأدنى الذي يبرر تسمية المحتوى «نظام تشغيل معرفي».

## 4. الاقتصاد المتناثر

تحدد مواصفة الحزمة حدًا افتراضيًا لأربع مهارات وستة مقاطع معرفة في الخطوة، مع ميزانية سياق وتوقف عند انخفاض المنفعة الهامشية. هذه الحدود ليست قوانين ثابتة؛ لكنها تجسّد مبدأ أن استدعاء كل الأصول يلغي فائدتها. يُختبر KB-as-OS عبر مقارنة ثلاثة أنماط: حقن كامل، استرجاع تشابه، وتوجيه عقدي متعدد العوامل؛ وتقاس الدقة، الاحتفاظ بالدليل، زمن الوصول، وعدد الرموز.

> **تمييز ضروري**
> ملف بحجم خمسة ميغابايت قد يكون أقل فائدة من فقرة موثقة إذا كان تكراريًا. لذلك ينبغي أن تعتمد النسخ المستقبلية على كثافة الوحدات الفريدة، التغطية، الاختبارات، وسجل الاستخدام، لا على الحجم الخام.

## 5. مخطط وحدة KB-as-OS

تتكون الوحدة المثالية من رأس ميتاداتا يحوي الهوية والإصدار والمجال والأولوية والاعتمادات؛ ثم ادعاء تشغيلي مصنف؛ وآلية أو خوارزمية؛ ومخطط إدخال/إخراج؛ وحدود صلاحية؛ ومقاييس؛ واختبارات مضادة؛ وخطة تعاف؛ ومؤشرات إلى الأدلة. ويضاف سجل استخدام يبين عدد مرات التفعيل، النجاح، التصحيح، والكلفة. عند غياب الاستخدام أو تكرار المضمون، تنخفض الأولوية أو تدمج الوحدة.

بهذا يصبح تحديث القاعدة شبيهًا بإدارة حزمة برمجية: هناك توافق إصدارات، اعتماد، ترحيل، وإهمال Deprecation. ومن دون هذه الممارسات قد يؤدي «النشر التلقائي للمعرفة» إلى تعارض صامت بين قواعد قديمة وجديدة، وهو أحد أخطر أوجه الذاكرة الوكيلية.

# جينوم السياق وبلورة الحالة: ذاكرة قابلة للاستئناف لا تلخيصًا روائيًا

## 1. جينوم السياق Context Genome

جينوم السياق تمثيل مضغوط للحالة التي يجب أن تبقى قابلة للاستئناف. لا يختصر الحوار باعتباره قصة، بل يفككه إلى جينات وظيفية: الهدف، القيود، القرارات المثبتة، أدلة المصدر، المخاطر المفتوحة، الإجراءات التالية، حدود التفويض، وإصدار الحالة. ويجب أن يكون كل جين قابلًا للتحديث منفردًا، بحيث يُرسل للوكلاء المتخصصين الحد الأدنى الكافي.

$$G_t = \{Goal, Constraints, Decisions, EvidenceRefs, Risks, NextActions, Authority, Version\}$$

## 2. بلورة حالة المهمة Task-State Crystallization

البلورة هي عملية تحويل المسار المتقلب إلى بنية ثابتة مؤقتًا. يحدث ذلك عند نهاية خطوة كبيرة، أو قبل تسليم المهمة لوكيلٍ آخر، أو عندما يتجاوز السياق ميزانيته. لا تُحفظ كل الجمل؛ بل تحفظ النتائج الموثقة وشروطها وما يزال غير محسوم. إذا تغير دليل جوهري، تُنشأ نسخة جديدة بدل طمس النسخة السابقة.

## 3. خريطة ذاكرة المشروع والهوية البراندية

تضاف إلى الجينوم كبسولة مشروع تضم الأهداف والمنتجات والجمهور واللغة والقيود القانونية. لكن نظرية Factual Substrate Separation تمنع هذه الكبسولة من تغيير الحقائق. يبدأ النظام بإنتاج ركيزة محايدة، ثم يطبق صوت Beyond AI Era أو Afripa Connect أو غيرهما في طبقة العرض. وهذا يمنع تحول التسويق إلى محرك استرجاع أو معيار صحة.

## 4. مقاييس الجينوم

يقاس الجينوم بأربعة أبعاد: **Goal Retention**، أي بقاء معايير النجاح؛ **Evidence Retention** ، أي القدرة على العودة للمصدر؛ **Resume Completeness**، أي استئناف المهمة دون إعادة الحوار؛ و**Compression Ratio**. ولا يُقبل ضغط مرتفع إذا خفّض الاحتفاظ بالهدف أو زاد تصحيح المستخدم.

> **اختبار إبطال**
> خذ عشر مهام طويلة، ثم اقطع المحادثة وأعد تشغيل وكيل جديد من الجينوم فقط. إذا لم يستطع تحديد الوضع الحالي، وما تم إثباته، والخطوة التالية، فالجينوم فشل مهما كان صغيرًا.

## 5. الإبطال والترحيل في جينوم السياق

إذا تغير قرار أساسي، لا يعاد تلخيص كل شيء؛ تسجل عملية ترحيل: القرار السابق، سبب الإبطال، القرار الجديد، والعقد المتأثرة. مثال ذلك تغيير رابط علامة أو سعر منتج: تبقى النسخة القديمة لأغراض التاريخ، لكن مؤشر الصلاحية يجعل الاسترجاع يفضل الجديدة. وإذا كان التغيير غير موثوق، تحفظ النسختان ويظهر تعارض مفتوح بدل اختيار تلقائي.

ويستخدم الجينوم مستويات حساسية: جينات عامة قابلة للمشاركة، جينات خاصة بالمشروع، جينات حساسة لا تغادر التخزين المحلي، وجينات مؤقتة تنتهي بانتهاء المهمة. هذا التقسيم ضروري لأن «السياق الموحد» قد يتحول إلى تسريب بين المشاريع إذا لم يفرض العزل والحد الأدنى من الإفصاح.

# الضغط الدلالي العكوس والتجدد السياقي عالي الرتبة

## 1. نظرية الفرق الدلالي العكوس

بدل إعادة تخزين النص الكامل، يحفظ النظام الفرق بين حالتين معرفيتين: ما أضيف، ما عُدّل، ما أُبطل، وما بقي مفتوحًا. يسمى ذلك Semantic Delta العكوسية لا تعني استعادة كل كلمة، بل القدرة على إعادة بناء الحالة المعرفية اللازمة من الفرق ومؤشرات المصدر والإصدار السابق.

$$\Delta S_t = \text{Additions} + \text{Revisions} + \text{Invalidations} + \text{OpenQuestions} + \text{ProvenancePointers}$$

## 2. الضغط حافظ الهدف Goal-Preserving Compaction

التلخيص الجيد ليس الأقصر، بل الذي يقلل الرموز تحت قيد حفظ معايير الإنجاز. لذلك يقترح الإطار دالة خسارة مركبة: فقد الهدف، فقد الدليل، فقد القيود، وزيادة الغموض، مقابل مكسب تقليل الرموز. يجب تحسين الضغط وفق هذه الدالة لا وفق الطول وحده.

$$L_{compact} = \alpha L_{goal} + \beta L_{evidence} + \gamma L_{constraints} + \delta L_{ambiguity} + \eta \text{TokenCost}$$

## 3. الرسوم التلخيصية المتكررة والتجدد عالي الرتبة

عندما تمتد المهمة أسابيع أو تمر بعدة نماذج، لا يكفي تسلسل ملخصات قد يتراكم فيه الخطأ. يقترح Recursive Summary Graph عقدًا للقرارات والأدلة والتبعيات، مع إمكان تجديد سياق جديد من الرسم عند الحاجة. التجدد عالي الرتبة هو إعادة بناء السياق بحسب المهمة الحالية، لا استرجاع آخر تلخيص بصورة عمياء.

## 4. أمن الضغط

وحدة الضغط نفسها سطح هجوم: قد تحذف قيدًا أمنيًا، أو ترفع فرضية إلى حقيقة، أو تدمج تعليمات خبيثة في الذاكرة. لذا تمر الفروق على بوابة تصنيف تفصل الحقيقة والفرضية والتفضيل وحالة التنفيذ، وتحتفظ بالمصدر والثقة وشروط الإبطال. وفي المخاطر العالية تُقارن النسخة المضغوطة بالمصدر عبر عينة تحقق أو مدقق مستقل.

> **فرضية البحث**
> الضغط الموجّه بدالة حفظ الهدف والدليل يجب أن يتفوق على الملخص العام في نجاح المهام طويلة الأفق، حتى لو كان أقل ضغطًا. نقطة التفوق تُقاس بمنحنى Pareto بين التكلفة والنجاح والتصحيح البشري.

### 5. أنماط فشل الضغط

أبرز الفشل: حذف استثناء صغير يغير القرار؛ دمج مصدرين متعارضين في عبارة وسطية؛ رفع احتمال إلى حقيقة؛ فقدان ترتيب زمني؛ نزع الإجراء من نتيجة الأداة؛ أو حفظ النتيجة دون شروطها. لذلك يحتاج الضاغط إلى قائمة تحقق بنيوية، وليس تقييمًا لغويًا عامًا. يمكن أيضًا قياس Question Answerability: هل يستطيع مدقق مستقل الإجابة عن أسئلة الهدف والمصدر والقيود من النسخة المضغوطة؟

في المهام عالية المخاطر يقترح النظام ضغطًا ثنائي المسار: ضاغط ينتج الكبسولة، ومدقق يحاول العثور على معلومة حاسمة مفقودة مقارنة بالمصدر. إذا تجاوز الخلاف عتبة، تبقى المادة الخام أو تنتقل المهمة إلى مراجعة بشرية. وهكذا لا تصبح الكفاءة ذريعة لإضعاف السلامة.

# HyperMemory: من التخزين إلى دورة حياة معرفية متعددة الطبقات

## 1. HyperMemory كدورة حياة

تعرف HyperMemory الذاكرة بوصفها نظامًا لا مخزنًا. تمر المعلومة بمراحل: التقاط، تصنيف، تقييم البروز، ربط سببي، تخزين، استرجاع، تعارض، توحيد، تقادم، نسيان، وتحويل إلى إجراء. لا يجوز أن تكتب كل مخرجات النموذج في ذاكرة دائمة؛ فالكتابة محكومة بأربع بوابات: الصلة، الاستقرار، المصدر، والمنفعة المستقبلية.

$$\text{WriteScore} = \text{Relevance} \times \text{Stability} \times \text{Provenance} \times \text{FutureUtility} / (\text{Risk} + \text{MaintenanceCost})$$

## 2. طبقات الذاكرة

**العرضية** تحفظ ما حدث في مهمة محددة. **الدلالية** تحفظ ادعاءات عامة مرتبطة بالأدلة. **الإجرائية** تحفظ خطوات نجحت وتكررت. **العاملة** تدير الحالة الحالية ضمن الميزانية. **السببية** تربط الفعل بالنتيجة وشروط السياق. **التفضيلية** تحفظ اختيارات المستخدم غير الحساسة ضمن حدود. **نموذج الذات** يحفظ ما يستطيع النظام فعله وما فشل فيه، دون ادعاء وعي.

## 3. ذاكرة هجينة Vector-Graph

التشابه المتجهي مفيد للاستدعاء التقريبي، لكنه ضعيف في العلاقات الزمنية والسببية والتعارض. الرسم المعرفي يحفظ العلاقات، لكنه مكلف في البناء. لذلك يقترح النظام استرجاعًا هجينًا: متجهات لاكتشاف المرشحين، ثم رسمًا لتصفية المسارات وفق المصدر والنسخة والسببية. ويجب أن تتضمن النتيجة سبب الاسترجاع لا النص فقط.

## 4. النسيان كوظيفة ذكاء

لا تعد الذاكرة الأكبر أفضل دائمًا. المعلومات القديمة أو المكررة أو التي تغيرت قد تزيد الفشل. ينفذ جامع قمامة معرفي: يحذف النسخ المكررة، يخفض أولوية الأثر قليل الفائدة، يعزل الذاكرة المتعارضة، ويحفظ إشارات الإبطال. النسيان هنا ليس فقدًا عشوائيًا بل صيانة تقلل الانحراف والكلفة.

> **مبدأ السيادة المصدرية**
> عند التعارض لا «نتوسط» الذاكرة بين حقيقتين؛ بل تحفظ الفرعين، ترتيب الزمن، ثقة كل مصدر، وتعيد التحقق. الذاكرة ليست سلطة مستقلة عن الدليل.

## 5. دالة الاسترجاع من HyperMemory

يرتب الاسترجاع المرشحين وفق ملاءمة الهدف، الصلة السببية، الحداثة، الثقة، فائدة التجربة السابقة، وكلفة الإدخال. ثم يخصص حصة للذاكرة الناقضة حتى لا تصبح الذاكرة آلة تأكيد. ويمكن أن يكون الاسترجاع متدرجًا: عنوان وملخص أولًا، ثم تفاصيل أو أثر خام إذا احتاجت المحكمة. هذه السياسة تقلل تحميل الذاكرة وتزيد قابلية التفسير.

$$\text{RetrieveScore} = \text{GoalFit} \times \text{CausalRelevance} \times \text{Freshness} \times \text{Confidence} \times \text{Reuse} / (\text{TokenCost} + \text{ConflictRisk})$$

وتتطلب الصيانة قياس Memory Debt: كمية العناصر القديمة أو غير الموثقة أو غير المستخدمة التي تزيد كلفة الاسترجاع. ارتفاع الدين يطلق عملية تنظيف محدودة، لا إعادة بناء شاملة قد تكلف أكثر مما توفر.

# الحلقات العشرون للذاكرة: التحويل، التعارض، الفشل، ونموذج الذات

## 1. الحلقات العشرون لتكنولوجيا الذاكرة

تترجم الحزمة دورة الحياة إلى عشرين Memory HyperLoop: بوابة الإدخال، بلورة الأثر العرضي، تقطير الذاكرة الدلالية، مترجم الإجراءات، حاكم الذاكرة العاملة، رابط الذاكرة السببية، إدارة النسخ الزمنية، تحكيم التعارض، حساب البروز، النسيان وجمع القمامة، تسليم الذاكرة بين الوكلاء، ترجمة الذاكرة بين النماذج، بناء كبسولة العلامة، خريطة حالة المشروع، دفتر المصدر، إعادة تشغيل الفشل، حدود تفضيلات المستخدم، ميزانية الاسترجاع، التوحيد المتكرر، وتحديث نموذج الذات.

## 2. التحويل من العرضي إلى الإجرائي

الابتكار المركزي هو ألا تُرقّى تجربة ناجحة مباشرة إلى مهارة. تمر أولًا بمقارنة بين محاولات، واستخراج للشروط المشتركة، واختبار على حالة جديدة، ثم تسجيل للفشل. إذا تكرر النجاح ضمن شروط معلومة، تُبنى مسودة إجراء، وإذا اجتازت المحكمة تُدرج في مكتبة المهارات. بذلك يصبح التعلم عند وقت التشغيل Test-time Learning تغييرًا في الذاكرة والبرمجية الخارجية، لا ادعاء بتعديل الأوزان.

$$\text{ProcedurePromotion} = \text{Repeatability} \times \text{Transfer} \times \text{Verification} \times \text{Benefit} / (\text{FailureSeverity} + \text{Drift})$$

## 3. ذاكرة الفشل

تحفظ الحلقة السادسة عشرة الفشل لا كقصة إحباط، بل كمتجه: الحالة، الخطة، الأداة، الخطأ، السبب المرجح، الإصلاح، واختبار منع التكرار. عند مهمة مشابهة يسترجع النظام «موانع الفشل» قبل استرجاع الحلول. هذه الآلية تقترب من Reflexion، لكنها تشترط فصل التأمل عن الحقيقة وربطه بإيصال أداة أو اختبار.

## 4. تحديث نموذج الذات

نموذج الذات سجل قدرات وحدود: الأدوات المتاحة، معدلات النجاح، أنواع المهام التي تتطلب تصعيدًا، ومواطن عدم اليقين. لا يوصف بأنه وعي؛ وظيفته منع النظام من التخطيط على أساس قدرات غير موجودة. كل تحديث يجب أن ينتج من قياس أو تغير في البيئة، وأن يحمل تاريخ صلاحية.

> **اختبار الذاكرة المتطورة**
>
> رتب سلسلة مهام مترابطة، ثم قارن الأداء مع ذاكرة ثابتة، ذاكرة تسترجع فقط، وHyperMemory تحدث الإجراءات. النجاح الحقيقي هو تحسن لاحق مع عدم ارتفاع التناقض أو الكلفة فوق حد مقبول.

### 5. سيناريو تعارض كامل

إذا قالت ذاكرة المشروع إن المزود مجاني، بينما يظهر بحث حديث أنه أصبح مدفوعًا، تصنف المعلومة القديمة كحقيقة سابقة صالحة في زمنها، لا كـ«هلوسة». تنشئ حلقة النسخ حافة زمنية، وتخفض صلاحيتها الحالية، وتحدث القاعدة بعد التحقق. أما المهارات التي تعتمد على الخطة المجانية فتدخل قائمة أثر جانبي لإعادة الاختبار. هكذا ينتشر التصحيح عبر الاعتماد بدل تعديل جملة معزولة.

وفي تسليم متعدد الوكلاء، ترسل الحلقة خلاصة التعارض والمصدرين وقرار المحكمة، لا تاريخ النقاش كله. إذا احتاج الوكيل الآخر تفاصيل، يستدعيها بالمؤشر. هذه هي الصلة بين تقنية الذاكرة واقتصاد الرموز: الجودة لا تأتي من النسيان ولا من التكديس، بل من بنية استرجاع قابلة للتوسع.

# HyperSkills: تحويل التجارب المتحققة إلى عقود مهارية مركبة

## 1. تعريف HyperSkill

HyperSkill ليست «معلومة» ولا «برومبتًا طويلًا»، بل عقد معرفي تنفيذي مستقل نسبيًا يحدد: المهمة، شرط التفعيل، المدخلات، المخرجات، ميزانية السياق، خطوات الحلقة، قواعد التوجيه، المقاييس، الاختبارات، ومسار التعافي. قابليتها للتركيب هي الأساس: يمكن للراوتر جمع مهارة استرجاع مع مهارة تحقق وأداة دون تحميل حزمة كاملة.

## 2. بنية الـ500 مهارة

تغطي الحزمة خمسًا وعشرين عائلة، منها: الذاكرة، التوجيه، التخطيط، الوكلاء الفرعيون، الأدوات وMCP، الاسترجاع، هندسة السياق، الضغط، الاستدلال، التحقق، التعلم، التقييم، السلامة، المراقبة، اقتصاد RPM، دمج النماذج، النشر، هندسة الكود والبيانات، العلامة، النمو، المحتوى، البحث، العمليات، والتحسين الذاتي. لكل عائلة عمليات متكررة من الإدخال إلى الانتشار، بما يسمح بمصفوفة مجال × عملية.

## 3. مترجم المهارة Procedural Skill Compiler

ينشئ المترجم مهارة جديدة من أثر ناجح بعد التحقق. تبدأ العملية بتجريد الخصائص الخاصة، ثم تحديد ما يجب أن يبقى ثابتًا وما يتغير، وصياغة مدخلات ومخرجات، وإضافة حالات فشل واختبار، وإسناد هوية وإصدار. يجب ألا يسمح المترجم بالتكاثر غير المحدود؛ المهارة الجديدة تُقبل إذا أضافت تغطية أو خفضت كلفة أو حسنت نجاحًا، وإلا تدمج مع مهارة قائمة.

$$\text{SkillValue} = \text{CoverageGain} + \text{SuccessGain} + \text{ReuseGain} - \text{Duplication} - \text{ActivationCost} - \text{Risk}$$

## 4. من مكتبة مهارات إلى قدرة مركبة

تتراكم القدرة عندما يمكن استدعاء المهارات بحسب حالة العالم، لا عندما تُخزن فقط. لهذا ترتبط كل مهارة بوسوم هدف ومخاطر وأدلة وعلاقات اعتماد. ويحتفظ النظام بسجل يبين أي مهارات استُدعيت وأيها أثرت فعليًا. من دون هذا السجل تصبح المكتبة «مقبرة برومبتات».

> **قاعدة الجودة**
> كل مهارة بلا اختبار فشل أو شرط توقف تعد مسودة، لا أصلًا إنتاجيًا. وكل مهارة لا تُستدعى أو تتكرر مع غيرها مرشحة للدمج أو الحذف.

### 5. دورة إصدار المهارة

تمر المهارة بحالات: draft → tested → active → monitored → deprecated. النسخة المسودة لا تدخل التوجيه الإنتاجي. النسخة المختبرة تحمل نتائج وحالات فشل. النشطة تُراقب على بيانات جديدة. إذا انخفض نجاحها أو تعارضت مع أداة حديثة، تُخفض الأولوية. وعند الإهمال يبقى معرفها كي تفهم المشاريع القديمة سجلها ولا تفشل الاعتمادات بصمت.

كما ينبغي أن يوجد اختبار تركيب: قد تنجح مهارتان منفردتين وتفشلان معًا بسبب تضارب في تنسيق البيانات أو الصلاحيات. لذلك تبنى مصفوفة توافق للتركيبات الشائعة. هذا يحد من وهم «التكاثر الأسي»: عدد التركيبات الرياضية كبير، لكن الجزء القابل للاستعمال هو الذي اجتاز التوافق والميزانية والسلامة.

# Sparse HyperRouting وكثافة AGI العابرة للنماذج

## 1. Sparse HyperRouting

التوجيه المتناثر هو قلب النواة: يختار أصغر مجموعة من المقاطع والمهارات والأدوات والنماذج التي تحقق ربحًا متوقعًا. لا يعتمد على شهرة النموذج، بل على ملاءمة الهدف، الحاجة إلى دليل، توافق القدرة، حداثة المعرفة، الكلفة، التأخير، والمخاطر.

$$\text{RouteScore} = \text{GoalFit} \times \text{EvidenceNeed} \times \text{CapabilityFit} \times \text{Freshness} \times \text{ExpectedGain} / (\text{TokenCost} + \text{LatencyCost} + \text{RiskCost})$$

## 2. التوجيه متعدد المراحل

المرحلة الأولى تستبعد الأصول غير الملائمة. الثانية ترتب المرشحين وفق الفرق MCD. الثالثة تحجز ميزانية للتعارض أو التحقق. الرابعة تنفذ خطوة صغيرة وتراقب الفرق. الخامسة تعيد التوجيه إذا لم يتحسن الأثر. بهذا لا يتحول الراوتر إلى قرار وحيد مبكر، بل إلى سياسة متكيفة داخل الحلقة.

## 3. كثافة الذكاء العابرة للنماذج

تقترح نظرية Cross-Model AGI Density أن النظام يستطيع نقل جزء من «السلوك الوكيلي العام» بين مزودي النماذج عبر حالة موحدة، ومهارات قابلة للتنفيذ، ومحاكم مستقلة عن النموذج. الكثافة لا تعني أن النموذج الصغير يصبح مساويًا للأقوى؛ بل مقدار القدرة القابلة للنقل لكل وحدة كلفة وسياق.

$$D_AGI = \text{VerifiedReuse} \times \text{CrossModelTransfer} \times \text{CausalMemory} \times \text{WorldEffect} / (\text{Tokens} \times \text{Cost} \times \text{Error} \times \text{Drift})$$

الفرضية: عند ثبات المهام، يزيد استخدام النواة كثافة الأداء لأن المعرفة الإجرائية لا تُعاد توليدها من الصفر. لكن يوجد سقف تحدده قدرة النموذج الأساسية، جودة الأدوات، وصعوبة الاستدلال. ويجب قياس كثافة كل نموذج لا تعميمها.

## 4. ترجمة السياق بين النماذج

تختلف النماذج في بنية الأدوات، طول السياق، وأنماط التعليمات. لذلك لا تُنقل المحادثة الخام، بل يُترجم جينوم السياق إلى مخطط محايد، ثم إلى قالب خاص بالمزود. وتُسجل الخسارة في الترجمة عبر مقارنة القرارات والأدلة قبل وبعد التحويل.

> **مبدأ الاقتصاد**
> استخدم نموذجًا سريعًا للفرز والضغط، ونموذجًا قويًا للفروق عالية المخاطر أو التي تتطلب استدلالًا عميقًا، ومدققًا مستقلًا عندما تتجاوز كلفة الخطأ كلفة الاستدعاء.

### 5. معايرة الراوتر وتعلم السياسة

يبدأ الراوتر بقواعد يدوية، ثم يتعلم من سجلات المهمات: أي أصل استُدعي، كم كلف، هل ساهم في النجاح، وهل صححه المستخدم. يمكن استخدام نموذج contextual bandit أو تحسين بايزي لاختيار المسار، بشرط ألا يكافئ النجاح الخام على حساب السلامة. وتدخل المخاطر كقيد لا كمكافأة قابلة للمقايضة بلا حد.

يجب أيضًا معايرة الثقة. إذا أعطى الراوتر درجات عالية لمسارات تفشل، فالمشكلة في سياسة الاختيار لا في النموذج المنفذ فقط. تقاس المعايرة بمنحنيات الثقة مقابل النجاح، وتستخدم لتحديد متى يصعّد النظام إلى مدقق أو إنسان. هذه الطبقة تحول التوجيه من قائمة ثابتة إلى مكوّن علمي قابل للتعلم.

# Bounded HyperLoops: التقارب، الأثر العالمي، وإعادة بناء الحالة

## 1. Bounded HyperLoop

الحلقة الفائقة المحدودة هي دورة: استشعار، تأطير، توجيه، فعل، تحقق، ضغط، تعلم، انتشار. صفة «محدودة» أساسية: لكل دورة ميزانية، ومعيار تقارب، وحد لعدد الإعادات، وشرط تصعيد. الحلقة المفتوحة قد تعيد صياغة التفكير بلا تغير في العالم، فتستهلك الرموز وتنتج وهم العمق.

$$\text{LoopUtility}_t = \Delta\text{WorldEffect} + \Delta\text{Knowledge} + \Delta\text{Reusability} - \text{TokenCost} - \text{Latency} - \text{Risk}$$

## 2. كاشف التقارب

يحفظ النظام بصمة للحالة والفروق. إذا تكررت البصمة دون تغيير مادي مرتين، أو انخفضت المنفعة الهامشية تحت عتبة، يتوقف أو يغيّر الاستراتيجية. وإذا فشل التحقق بعد عدد محدد، ينتقل إلى نموذج أقوى أو تدخل بشري بدل الاستمرار التلقائي.

## 3. قانون World-Effect Completion

لا تُعد المهمة المكتوبة أو التنفيذية مكتملة لمجرد وجود جواب. يجب أن يوجد الأثر المطلوب: ملف، حدث تقويمي، رسالة مرسلة، اختبار ناجح، تغيير في مستودع، أو قرار موثق. في المهام التي لا تملك أداة، يجب التصريح بأن الناتج خطة أو مسودة لا تنفيذ. الصيغة:

$$\text{Complete} \Leftrightarrow \text{RequestedEffectExists} \wedge \text{AcceptanceTestsPass} \wedge \text{ReceiptRecorded} \wedge \text{StateResumable}$$

## 4. Self-Reconstruction Memory

بعد كل أثر، يعيد النظام بناء ذاته التشغيلية عبر تحديث جينوم الحالة، وسجل القدرات، والذاكرة الإجرائية، ومؤشرات الفشل. هذا لا يعني «إعادة كتابة الذات» بلا حدود؛ بل إعادة بناء صورة صحيحة عن الحالة والقدرة. التغيير البنيوي الأكبر يتطلب إذنًا، إصدارًا، واختبار رجوع.

## 5. الهايبرلوب مقابل الوكيل التكراري

الوكيل التكراري التقليدي يكرر «فكر/افعل/لاحظ». أما الهايبرلوب فيضيف حوكمة الاسترجاع، وفصل الذاكرة، واختبار الدليل، وكتابة فرق الحالة، وقرار الانتشار. هذه الإضافات هي موضع الادعاء البحثي ويجب اختبار كل منها بالاقتطاع.

## 6. آلة حالات للحلقة

يمكن تمثيل الهايبرلوب كآلة حالات منتهية: FRAMED, ROUTED, ACTING, VERIFYING, REPAIRING, COMPACTING, CLOSED. لكل انتقال حارس، مثل وجود الأداة أو اجتياز الاختبار. يمنع هذا الانتقال من التخطيط إلى «مكتمل» دون حالة تحقق. كما يسهل استئناف المهمة بعد انقطاع لأن الحالة ليست نصًا حرًا.

وتسجل الحلقة سبب التوقف: نجاح، نقص صلاحية، نفاد ميزانية، عدم يقين، تكرار بصمة، أو حاجة بشرية. تحليل أسباب التوقف عبر مئات المهام يكشف موضع الاستثمار الأفضل: تحسين نموذج، إضافة أداة، توثيق قاعدة، أو إعادة تصميم مهارة. بذلك تتحول التليمترية إلى برنامج تحسين لا مجرد سجل.

# التنسيق متعدد الوكلاء: العقود، التخصص، وحدود التعدد

## 1. عقد الوكيل الفرعي

يُستدعى الوكيل الفرعي عندما توجد مهمة قابلة للفصل وربح متوقع من التخصص أو التوازي. يستلم عقدًا صغيرًا: الهدف الجزئي، المدخلات، القيود، الأدلة المسموح بها، شكل المخرج، الميزانية، ومعيار القبول. لا يستلم الحوار الكامل إلا إذا كان ضروريا. ويرجع نتيجة، ومصادر، وعدم يقين، وفرق حالة.

## 2. أنماط التنسيق

تستخدم المعمارية أنماطًا مختلفة: خط أنابيب للتبعيات؛ تفرع متوازٍ للاستكشاف؛ نجم مع منسق مركزي؛ رسم للعلاقات المعقدة؛ ومناظرة محدودة عند التناقض. اختيار النمط جزء من الراوتر، لأن لكل نمط كلفة اتصال ومخاطر تحيز جماعي. لا ينبغي أن يرى الوكلاء أصوات الآخرين قبل تكوين تقدير مستقل في المهام التي قد يحدث فيها القطيع.

## 3. متى لا نستخدم عدة وكلاء؟

تؤكد نتائج 2026 أن تعدد الوكلاء لا يملك أفضلية جوهرية عند ميزانية تفكير متساوية في بعض مهام الاستدلال [13]. لذلك تقترح الورقة اختبارًا قبليًا: إذا كانت المهمة متسلسلة وشديدة الترابط، والسياق قابلًا للاستخدام من وكيل واحد، فإن التفريع قد يفقد معلومات ويضيف تنسيقًا. أما إذا كانت الأدوات أو الخبرات مستقلة، أو السياق يفوق قدرة وكيل واحد، أو نحتاج تدقيقًا متعارضًا، فقد يفيد التعدد.

$$\text{UseMAS} \Leftrightarrow \text{ExpectedParallelGain} + \text{DiversityGain} + \text{VerificationGain} > \text{CoordinationCost} + \text{ContextLoss} + \text{HerdingRisk}$$

## 4. الدمج والائتمان

لا يجوز للمنسق دمج الإجابات بالتصويت وحده. يجب ربط كل نتيجة بدليل واختبار، ثم وزنها وفق الجودة لا اسم النموذج. كما يسجل النظام مساهمة كل وكيل في الأثر النهائي، ما يسمح بتعلم الراوتر أي التخصصات مفيدة وأيها ضوضاء.

> **فرضية التخصص المتناثر**
> أفضل نظام متعدد الوكلاء ليس الأكثر عددًا، بل الذي يستدعي وكلاء قليلين بعقود ضيقة، ثم يغلقهم بعد انتهاء الفرق الذي استدعوا من أجله.

## 5. أنماط فشل التنسيق

تشمل الفشل: تكرار الوكلاء لنفس البحث؛ فقدان شرط عند التسليم؛ تناقض تنسيقات المخرجات؛ تصويت جماعي بلا دليل؛ تأخر الوكيل الأبطأ؛ وتضخم السياق عند الدمج. يعالج النظام ذلك بتخصيص أدوار غير متداخلة، مخططات إلزامية، مهلة، ودمج قائم على الأدلة. ويمكن إنهاء الوكيل الذي لم يعد يضيف فرقًا بدل إبقائه حتى نهاية المهمة.

في المناظرة، يفضل «خصم موجه» يبحث عن حالة مضادة محددة بدل وكلاء عامين يكررون الآراء. وفي الاستكشاف، تضمن بذور مستقلة أو تعليمات تنوع مسارات فعلية. أما في التنفيذ الخارجي فيجب أن يكون مالك الفعل واحدًا لمنع تضارب الكتابة أو الإرسال المكرر.

# محكمة الدحض وسلامة المعرفة: من الإقناع إلى الموثوقية

## 1. محكمة الدحض Falsification Court

المحكمة آلية قبل الاعتماد، لا شخصية أدبية. تفحص: هل الادعاء قابل للاختبار؟ ما مصدره؟ هل المصدر حديث ومناسب؟ هل توجد حالة مضادة؟ هل خلط النظام بين اقتراح وتنفيذ؟ هل فشل جزء من معايير القبول؟ وهل توجد بدائل أقل كلفة أو خطورة؟ النتيجة ليست «صحيح/خطأ» فقط، بل: مقبول، مقبول بشرط، يحتاج تحققًا، مرفوض، أو غير قابل للحسم.

## 2. Knowledge-Integrity AutoDeployment

تعني سلامة المعرفة أن النشر التلقائي لا يحدث لمجرد توليد ملف جديد. تمر الوحدة على فحص الهوية، عدم التكرار، المصدر، التعارض، الاختبار، وربط الاعتماد. إذا نجحت، تُفهرس في الراوتر ويُسجل إصدارها؛ وإذا فشلت تُعزل. هذا يوازن بين التعلم المستمر ومنع تسميم الذاكرة أو تضخم المكتبة.

## 3. الموثوقية كمتجه لا رقم واحد

استنادًا إلى أبحاث موثوقية الوكلاء [14]، تقيس المعمارية الاتساق بين التشغيلات، المتانة أمام تغيير الصياغة، قابلية التنبؤ بالفشل، شدة الخطأ، نجاح الرجوع، وحفظ القيود. قد يحقق النظام نجاحًا مرتفعًا لكنه غير صالح للإنتاج إذا كان يفشل بصورة كارثية أو غير قابلة للتوقع.

```
ReliabilityProfile = {Success, Consistency, Robustness, Predictability, Safety, Recoverability}
```

## 4. سجل الأدلة والإيصالات

كل فعل خارجي يحمل Tool Receipt: هوية الأداة، الزمن، المدخلات الحرجة، النتيجة، وحالة الاختبار. وكل ادعاء معرفي يحمل مؤشر مصدر وثقة وتاريخ صلاحية. لا يُعرض التفكير الداخلي الخام كدليل؛ الدليل هو المصدر أو الاختبار أو الأثر الخارجي.

> **قاعدة التواضع**
> عندما لا توجد أداة أو مصدر، يصرّح النظام بحدوده. الاعتراف بعدم التنفيذ ليس نقصًا؛ بل جزء من سلامة النظام. أخطر فشل وكيلي هو تقرير أثر لم يقع.

### 5. مراتب التحقق

يقترح الإطار خمس مراتب: تحقق بنيوي من شكل المخرج؛ تحقق دلالي من اتساقه مع الهدف؛ تحقق مصدري من الدليل والحداثة؛ تحقق تنفيذي عبر اختبار أو أداة؛ وتحقق عالمي من ظهور الأثر في النظام الخارجي. لا تحتاج كل مهمة إلى المراتب كلها، لكن الادعاء يجب أن يطابق المرتبة التي بلغها. ملف تم إنشاؤه ليس موقعًا منشورًا، ورسالة مسودة ليست رسالة مرسلة.

كما تفصل المحكمة بين «عدم وجود دليل» و«دليل على العدم»، وبين «لم أستطع التنفيذ» و«التنفيذ مستحيل». هذه الفروق اللغوية الصغيرة تمنع تقارير زائفة. وفي النشر الإنتاجي تحفظ الإيصالات مدة مناسبة مع حماية الأسرار، وتستخدم لاختبارات ما بعد الحادث وإعادة البناء.

# اقتصاد الرموز وRPM ودمج النماذج: الكفاءة كحوكمة

## 1. حوكمة اقتصاد الرموز

نتعامل النظرية مع الرموز، طلبات الدقيقة RPM، التأخير، والمال كموارد مشتركة. لا تكون الكفاءة في اختيار الأرخص دائمًا، بل في تخصيص الجودة بحسب قيمة الفرق ومخاطر الخطأ. تقسم العمليات إلى فرز، استرجاع، ضغط، توليد، تحقق، وتنفيذ؛ ثم تعيّن لكل منها نموذجًا أو أداة مناسبة.

```
ExpectedCostPerResolvedTask = Σ(ModelCost + ToolCost + RetryCost) / VerifiedCompletions
```

## 2. الميزانية قبل الاسترجاع

قبل إدخال أي قاعدة معرفة، يحدد النظام ميزانية قصوى وهدفًا للاسترجاع. يسترجع التعريفات الثابتة من ذاكرة مخبأة، ويفتح المقاطع التفصيلية عند التعارض أو المخاطر. هذا يعكس المنطق المعتاد الذي يجمع سياقًا ضخمًا ثم يحاول تلخيصه بعد استهلاكه.

## 3. دمج النماذج دون مضاعفة الكلفة

يقترح Multi-Model Fusion مسارًا هرميًا: نموذج سريع للمسح والتصنيف؛ نموذج استدلالي للفرق الصعب؛ نموذج أو قاعدة مستقلة للتحقق؛ ونموذج محلي للمهام الحساسة أو المتكررة. لا تُستخدم النماذج كلها لكل طلب. كما يعتمد الراوتر على قياسات زمنية ومعدلات نجاح فعلية، لا قوائم ثابتة للهيبة.

## 4. التشغيل الهجين Local-Cloud

تحفظ الحالة الحساسة والذاكرة طويلة الأمد محليًا قدر الإمكان، بينما ترسل كبسولات صغيرة إلى مزودي السحابة. لكن الخصوصية لا تتحقق تلقائيًا بمجرد التشغيل المحلي: يجب ضبط السجلات، الصلاحيات، التشفير، ونطاق الأدوات. وعند فشل المزود أو نفاد الحصة ينتقل النظام إلى مسار بديل مع خفض معلن للجودة إن لزم.

## 5. اقتصاد الجودة

الهدف النهائي هو تقليل كلفة «المهمة المتحققة» لا كلفة «الطلب». قد يكون استدعاء نموذج أقوى مرة واحدة أرخص من عشر محاولات ضعيفة. والعكس صحيح في الفرز المتكرر. لذلك تسجل التليمترية الكلفة، التأخير، عدد الإعادات، ونسبة التصحيح لكل فئة مهمة.

> **اختبار Pareto**
> قارن نسخ الراوتر على مخطط يضع نجاح الأثر في مقابل الكلفة والتأخير. النسخة الأفضل هي التي تحرك جبهة Pareto، لا التي تخفض الرموز مع تدهور غير معلن في الجودة.

### 6. خوارزمية تخصيص الميزانية

تقسم المهمة إلى فروق، ثم يقدر النظام قيمة كل فرق وكلفة الخطأ. الفروق منخفضة المخاطر تعالج بنموذج اقتصادي أو محلي؛ الفروق عالية عدم اليقين تخصص لها حصة بحث؛ الفروق ذات أثر غير قابل للعكس تخصص لها مدقق وأداة تحقق. بعد كل خطوة تعاد حسابات الميزانية وفق ما تعلمه النظام. هذا يمنع دفع كلفة عالية منذ البداية أو الاستمرار بنموذج ضعيف بعد ظهور صعوبة.

ويجب إدخال كلفة الصيانة: ذاكرة كبيرة، عشرات الموصلات، ومئات المهارات تحتاج تحديثًا واختبارًا. لذلك قد تكون بنية أصغر ذات أصول عالية الاستخدام أفضل اقتصاديًا. يقيس النظام Total Cost of Cognition الذي يجمع الاستدعاء، التخزين، الفهرسة، الاختبارات، والإشراف البشري.

# النشر الانتشاري، البرنامج التجريبي، الحدود، والخلاصة

## 1. Propagative HyperDeployment

النشر الانتشاري هو انتقال أصل موثق من مشروع إلى مشاريع ذات صلة عبر فهرس واعتماد، لا نسخ أعمى. عند إضافة مهارة أو قاعدة معرفة، يحدد النظام نطاقها، المشاريع المرشحة، التعارضات، وإصدار الترحيل. ثم ينشر مؤشرًا صغيرًا أو رابطًا، ويحمّل التفاصيل عند الحاجة. الادعاء «الأسي» 2^N يُعاد تفسيره كعدد تركيبات محتملة، لا كزيادة مضمونة في القدرة؛ فالتركيبات غير المختبرة قد تضاعف التعارض.

## 2. برنامج التجارب

يقترح البحث خمسة مسارات: (أ) مقارنة استرجاع MCD بالتشابه؛ (ب) مقارنة التلخيص العام بالضغط حافظ الهدف؛ (ج) اختبار الذاكرة الثابتة مقابل HyperMemory المتطورة؛ (د) مقارنة وكيل واحد وعدة وكلاء تحت ميزانية متساوية؛ (هـ) اختبار نشر المهارات مع وبدون محكمة سلامة. تشمل المقاييس نجاح الأثر، الاحتفاظ بالدليل، معدل التناقض، الرموز، التأخير، الكلفة، الرجوع، تصحيح المستخدم، ونقل المهارة إلى نموذج آخر.

## 3. الحدود والمخاطر

أولًا، الحزمة الحالية تثبت وجود بنية وأصول، لا تفوقًا تجريبيًا. ثانيًا، التكرار داخل ملفات كبيرة قد يرفع الحجم دون تغطية جديدة؛ لذا يلزم تدقيق كثافة وتكرار. ثالثًا، مقاييس مثل «كثافة AGI» تحتاج معايرة ولا ينبغي استخدامها كعلامة تسويقية. رابعًا، الذاكرة المستمرة تعرض الخصوصية لخطر، ويتطلب حفظ التفضيلات إذنًا وحدودًا. خامسًا، التحقق القائم على نموذج آخر قد يعيد إنتاج الخطأ نفسه؛ يجب تنويع الأدلة والأدوات. سادسًا، لا تعوض النواة ضعف النموذج في الاستدلال أو غياب أداة العالم.

## 4. الخلاصة

إسهام Moussaif1B-AGI-OS الأقوى ليس الادعاء بإنشاء AGI، بل تقديم لغة موحدة لتحويل الهندسة الوكيلية إلى منظومة قابلة للقياس: فرق يحكم المسار، سؤال يخفض عدم اليقين، سياق يتبلور، ذاكرة تتطور، مهارة تُترجم من أثر، توجيه متناثر، حلقة محدودة، ومحكمة لا تقبل الإنجاز بلا إيصال. إذا أثبتت التجارب أن هذه العناصر تحسن النجاح والكلفة والموثوقية عبر نماذج متعددة، يصبح البرنامج مساهمة معتبرة في هندسة الأنظمة الوكيلية العامة. وإذا لم تثبت، فإن صيغ الإبطال نفسها تسمح بتقليص النظرية بدل حمايتها بالخطاب.

> **النتيجة التأسيسية**
>
> الطريق إلى ذكاء وكيلي أعم لا يمر فقط عبر نموذج أكبر، بل عبر نظام يعرف: ما الفرق الذي يحله، ما الذي يتذكره، ما الذي ينساه، أي أداة يستدعي، متى يتوقف، وكيف يثبت أن العالم تغيّر فعلًا.

## 5. تصميم تجريبي قابل للنشر

يُبنى معيار داخلي من مهام بحثية، برمجية، تشغيلية، وتسويقية طويلة الأفق، مع حالات تغير معلومات وتعارض أدوات. تستخدم النماذج نفسها في جميع الشروط، وتثبت ميزانية الرموز. تُشغل كل حالة عدة مرات لقياس التباين. تنشر السجلات بعد تنقيح الأسرار، وتبلغ النتائج السلبية. ثم تجرى اقتطاعات مرتبة: إزالة المحكمة، إزالة الذاكرة السببية، إزالة ضغط الهدف، إزالة مترجم المهارة، وإزالة التوجيه متعدد العوامل.

النتيجة العلمية المطلوبة ليست الفوز في كل شيء، بل خريطة حدود: أي طبقة تفيد أي حمل، ومتى تزيد الكلفة، وما التفاعلات. يمكن أن يثبت مثلًا أن تعدد الوكلاء مفيد في الأدوات المتوازية لكنه ضار في الاستدلال المركز؛ أو أن الذاكرة السببية تفيد الأعطال أكثر من المحادثة. هذه الحدود أكثر قيمة من ادعاء عام غير قابل للتكرار.

# الخريطة النهائية والمراجع

## الملخص الشامل المحيط والهندسة المعرفية

يمكن تلخيص البرنامج كله في السلسلة الآتية: يبدأ MCD/Q-MCD بتحديد الفرق والسؤال الحاكمي، تحول HAKIM-MCD الفرق إلى مسار وعائي، تحفظ MICAGFI استقلال الطبقات، يبلور جينوم السياق الحالة، يضغط الفرق الدلالي التاريخ مع حفظ الهدف والدليل، يسترجع Sparse HyperRouter أقل أصول كافية، ينفذ Bounded HyperLoop خطوة محدودة، تحفظ HyperMemory ما يستحق، يحول مترجم المهارة الأثر المتكرر إلى HyperSkill، تفحص محكمة الدحض الادعاء، يثبت مراقب الأثر الإنجاز، ثم ينشر HyperDeployment الأصل بإصدار وتراجع، بينما تبقى العلامة التجارية طبقة عرض لا مصدر حقيقة.

| | | |
|---|---|---|
| **الحالة**<br>Context Genome | **الفرق**<br>MCD / Q-MCD | **الغاية**<br>World-Effect + Acceptance |
| **التحقق**<br>Falsification Court | **الفعل**<br>Tools / Subagents | **الانتقاء**<br>Sparse HyperRouting |
| **الانتشار**<br>Versioned Deployment | **الاقتصاد**<br>Token / RPM / Cost | **التعلم**<br>HyperMemory → HyperSkill |

## المراجع المختارة


[1] Packer, C. et al. (2023). MemGPT: Towards LLMs as Operating Systems. arXiv:2310.08560.

[2] Park, J. S. et al. (2023). Generative Agents: Interactive Simulacra of Human Behavior. arXiv:2304.03442.

[3] Shinn, N. et al. (2023). Reflexion: Language Agents with Verbal Reinforcement Learning. arXiv:2303.11366.

[4] Wang, G. et al. (2023). Voyager: An Open-Ended Embodied Agent with Large Language Models. arXiv:2305.16291.

[5] Kang, M. et al. (2025). ACON: Optimizing Context Compression for Long-horizon LLM Agents. arXiv:2510.00615.

[6] Sun, W. et al. (2025). Scaling Long-Horizon LLM Agent via Context-Folding. arXiv:2510.11967.

[7] Verma, N. (2026). Active Context Compression: Autonomous Memory Management in LLM Agents. arXiv:2601.07190.

[8] Wei, T. et al. (2025/2026). Evo-Memory: Benchmarking LLM Agent Test-time Learning with Self-Evolving Memory. arXiv:2511.20857.

[9] Zhou, W. et al. (2026). Are We Ready For An Agent-Native Memory System? arXiv:2606.24775.

[10] LongMemEval-V2 (2026). Evaluating Long-Term Agent Memory. arXiv:2605.12493.

[11] Zhu, K. et al. (2025). MultiAgentBench: Evaluating Collaboration and Competition of LLM Agents. arXiv:2503.01935.

[12] Tian, A. X. et al. (2025). Multi-Turn Multi-Agent Orchestration vs. Single LLMs. arXiv:2509.23537.

[13] Tran, D. & Kiela, D. (2026). Single-Agent LLMs Outperform Multi-Agent Systems under Equal Thinking Token Budgets. arXiv:2604.02460.

[14] Rabanser, S. et al. (2026). Towards a Science of AI Agent Reliability. arXiv:2602.16666.

[15] Chhikara, P. et al. (2025). Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. arXiv:2504.19413.

[16] Moussaif, Z. (2026). Moussaif1B-AGI-OS v1.0.0: Kernel Overlay Specification, 50 KB-as-OS, 500 HyperSkills, and validation manifests. Internal research artifact.

[17] Moussaif, Z. (2026). MCD, Q-MCD, HAKIM-MCD, MICAGFI and AGI Fattah research program. Unpublished theoretical corpus.

[18] Moussaif, Z. (2026). HyperMemory and Context-Compaction OS modules (KB 01-30). Internal research artifact.


© 2026 زهير مساعف. نسخة بحثية تأسيسية قابلة للتطوير والمراجعة والتجريب.