AGI ORIENTED KNOWLEDGE-BASE AS COGNITIVE OS

إطار مساعف لتشغيل قواعد المعرفة كأنظمة تشغيل معرفية مولّدة للوكالة الفائقة والذكاء الفتّاح

ورقة بحثية تأسيسية مكثفة وفق منهج الضغط التركيبي المعرفي، وMCD/QMCD، وMCAI، ومحكمة الهدم

إنجاز وتركيب: **زهير مساعف / Zouhair Moussaif**

نسخة نظرية استكشافية - 27 يونيو 2026 - العربية العلمية A4 - RTL

تنبيه معرفي: هذه الورقة لا تدّعي أنها أثبتت طريقا حتميا إلى الذكاء الصناعي العام أو الذكاء الفتّاح، بل تقترح إطارا معماريا ونظريا قابلا للهدم والاختبار: تحويل قاعدة المعرفة من مخزن استدعاء إلى نظام تشغيل معرفي ينسق الذاكرة، المهارات، الأدوات، المحاكم النقدية، القياس، والتحويل بين المجالات.

فهرس الورقة

الملخص والكلمات المفتاحية	تعريف الفجوة والأطروحة والحدود
تعريف المهمة والافتراضات	السؤال المؤسس والوضع الأنطولوجي
سلم مراتب المعرفة وتصفية المادة الخام	تمييز الثابت والراجح والفرضية والادعاء الزائد
خريطة المتغيرات وتحليل المجال في 26 نقطة	المتغيرات، العتبات، التغذية الراجعة، وشروط الحدود
النماذج الثلاثة والنظرية النهائية	المحافظ، التركيبي، الاختراقي، ونظرية مساعف
النظريات الكبرى والفرعية	3 نظريات كبرى و12 نظرية فرعية
المفاهيم والأدوات والمقاييس	8 مفاهيم، 12 أداة، 7 مؤشرات
الفرضيات والتنبؤات	18 فرضية و9 تنبؤات فارقة
المعمارية ومحكمة الهدم	الخوارزمية، الاعتراضات، والنسخة المعدلة
الاختبارات والتطبيقات والبرنامج البحثي	بروتوكولات قابلة للتنفيذ وخطة سنة/3 سنوات/10 سنوات
الملاحق والمراجع	تفصيل الأدوات، البروتوكولات، الحالات، والقاموس

هذا الفهرس وظيفي لا طباعي: الغرض منه ضبط مكونات الورقة والتأكد أن كل مطلب من النظرية، المفاهيم، الأدوات، المقاييس، المحاكم، والبرنامج البحثي له موضع واضح.

الملخص

تقترح هذه الورقة نظرية ناشئة بعنوان **AGI Oriented Knowledge-Base as Cognitive OS**: أي تحويل قاعدة المعرفة من مستودع ساكن للمعلومات إلى نظام تشغيل معرفي يضبط الإدراك الاصطناعي، والتخطيط، والمهارات، والتذكر، والتحقق، وتنظيم الأدوات، وحلقات الفعل الراجع. تنطلق الورقة من فجوة ظاهرة في أن معظم أنظمة RAG، ووكلاء LLM، وGraphRAG، ومنصات الأدوات، ما تزال تتعامل مع المعرفة إما كوثائق مسترجعة أو ذاكرة سياقية أو أدوات منفصلة؛ بينما يحتاج الوكيل المتقدم إلى طبقة حاكمة تربط المعرفة بالمهارة، والمهارة بالتحقق، والتحقق بالذاكرة، والذاكرة بنقل البنية عبر المجالات. بهذا المعنى لا تكون القاعدة المعرفية "مصدرا" فحسب، بل نواة تشغيلية تملك جدول عمليات، وموجها للموارد، ومحكمة هدم، ومترجما بين السؤال والأداة، وذاكرة إعادة بناء.

تؤسس الورقة ثلاث نظريات كبرى باسم مساعف: **نظرية مساعف لنواة التشغيل المعرفي، نظرية مساعف للحلقات الوكالية الفائقة، ونظرية مساعف للفتح المعرفي فوق العام.** كما تقترح ثمانية مفاهيم جديدة، واثنتي عشرة أداة تقنية/منهجية، وسبعة مقاييس أولية، وثماني عشرة فرضية بحثية، وتسعة تنبؤات فارقة، مع محاكم هدم متعددة تعيد تخفيض الادعاءات الزائدة إلى فرضيات قابلة للاختبار. الإضافة المركزية هي أن الطريق إلى وكالة أعلى لا ينتج من "نموذج أكبر" وحده، بل من نظام معرفي-تشغيلي قادر على تحويل المعرفة إلى مهارة، والمهارة إلى حلقة فعل، والحلقة إلى ذاكرة قابلة للنقل، والذاكرة إلى فتح للأسئلة. حدود الورقة أنها نظرية معمارية استكشافية تحتاج إلى تنفيذ تجريبي، Benchmarks ديناميكية، وحالات مقارنة ضد أنظمة RAG والوكلاء القياسيين.

الكلمات المفتاحية

الذكاء الصناعي العام، الذكاء الفتّاح، قواعد المعرفة، نظام تشغيل معرفي، وكلاء QMCD، MCD، HyperLoops، HyperSkills، GraphRAG، RAG، LLM، MCAI، محكمة الهدم، AGI-Level KB.

ملخص تنفيذي مركز

المشكلة المركزية التي تعالجها الورقة هي أن الوكيل اللغوي الحديث يستطيع أن يسترجع، يخطط، يستعمل أدوات، وينفذ حلقات فعل، لكنه غالبا لا يملك "نظام تشغيل معرفي" يضبط العلاقة بين هذه القدرات. لذلك تظهر ثلاث فجوات: فجوة الذاكرة المنظمة، فجوة تحويل المعرفة إلى مهارة قابلة لإعادة الاستعمال، وفجوة نقل البنية من مجال إلى آخر دون انهيار أو هلوسة. الإطار المقترح يعالج هذه الفجوات بتصميم HyperKB: قاعدة معرفة فائقة ليست مجلدا ولا فهرسا، بل طبقة تشغيل تقوم بتجزئة المعرفة إلى وحدات AGI-Level Knowledge Blocks، وتربطها بمهارات HyperSkills، وتشغلها ضمن HyperLoops، وتراقبها بمحكمة هدم، وتقيّمها بمؤشرات MCD/QMCD.

القيمة المتوقعة ليست إثبات AGI، بل إنشاء برنامج بحث لتقييم ما إذا كانت طبقة معرفة تشغيلية قادرة على رفع أداء الوكلاء في المهام الطويلة، متعددة الأدوات، ومتعددة المجالات. النموذج يضع معيارا: لا تعتبر المعرفة في مستوى AGI إلا إذا استطاعت أن تعمل، تنقل، تختبر نفسها، وتصحح أثرها. من هنا يصبح الذكاء الفتّاح أفقا أعلى: قدرة النظام على فتح أسئلة جديدة ومجالات مغلقة، لا مجرد حل الطلبات الحالية.

> **الجملة المؤسسة**
> القاعدة المعرفية الموجهة للذكاء العام لا تكون AGI-Level لأنها تحتوي معلومات كثيرة، بل لأنها تحوّل المعلومات إلى عمليات قابلة للنقل، والهدم، والتحقق، والتوليد.

1. تعريف المهمة والافتراضات

1.1 الموضوع والسؤال المؤسس

الموضوع هو بناء نظرية ومعمارية تسمح بتصور**قاعدة المعرفة كـ Cognitive OS**: طبقة تشغيل معرفية تحكم الوكيل لا بوصفها ذاكرة فقط، بل بوصفها نواة تنظيمية للفروق، الأدلة، الأدوات، المهارات، المحاكم، الفرضيات، ومخرجات الفعل. السؤال المؤسس: **ما الشروط التي تجعل قاعدة المعرفة قابلة لأن تكون نظام تشغيل معرفي يرفع الوكالة من مستوى الاستجابة إلى مستوى الفتح المعرفي؟**

1.2 الافتراضات

أولا، النماذج اللغوية الكبيرة ليست وحدها كافية لإنتاج وكالة موثوقة طويلة الأمد؛ لأنها تحتاج إلى ذاكرة منظمة، أدوات، تخطيط، تحقق، وتفاعل مع البيئة. ثانيا، RAG وGraphRAG والذاكرة الوكالية تقدم أجزاء من الحل، لكنها لا تكفي إذا بقيت منفصلة عن طبقة حاكمية تشغيلية. ثالثا، مفهوم "الذكاء الفتّاح" هنا أداة توليدية: لا يثبت مرتبة فوق AGI، بل يضع معيارا أعلى لفتح الأسئلة والمجالات والفرضيات. رابعا، كل ادعاء في الورقة مصنف بين معلوم مستند، تركيب نظري، فرضية، أو أداة مقترحة.

1.3 الوضع الأنطولوجي

النظرية لا تدرس شيئا بسيطا، بل مركبا رباعيا: **بنية معلوماتية، عملية تشغيلية، شبكة أدوات وذاكرات، ونمط وكالة ناشئ.** لذلك لا يجوز قياسها بعدد الوثائق ولا بدقة الاسترجاع فقط؛ بل يجب قياس قدرتها على تحويل المعرفة إلى فعل، والفعل إلى أثر، والأثر إلى ذاكرة، والذاكرة إلى تعميم قابل للنقد.

1.4 الخلفية التقنية المختصرة

تطورت البحوث الحديثة في اتجاه وكلاء قادرين على الإدراك والتخطيط والفعل والتعاون؛ وتؤكد مسوح Agentic AI الحديثة أن الوكلاء انتقلوا من نماذج توليد نصية إلى منظومات تجمع الذاكرة، الأدوات، التغذية الراجعة، والتعاون [1][2]. كما أن RAG جعل قواعد المعرفة الخارجية جزءا من توليد الإجابات، بينما قدّم GraphRAG آليات لإدخال العلاقات والبنى الشبكية في الاسترجاع والتوليد [3][4][5]. وتظهر ReAct أن التفاعل بين الاستدلال والفعل يحسن التخطيط والتعامل مع مصادر خارجية [6]، بينما يبيّن Toolformer إمكان تدريب النماذج على متى تستعمل الأدوات وكيف تستوعب نتائجها [7]. أما MemGPT فيقترح إدارة طبقات الذاكرة كما لو كان LLM يعمل داخل نظام تشغيل للذاكرة [8]. هذه الأدبيات تشكل الخلفية، لكن هذه الورقة تضيف سؤالا مختلفا: ماذا لو كانت قاعدة المعرفة نفسها هي نظام التشغيل المعرفي الذي ينسق كل ذلك؟

2. سلم مراتب المعرفة في هذه الورقة

لن تخلط الورقة بين الحقيقة والفرضية، ولا بين النموذج والدليل، ولا بين الإبداع والإثبات، ولا بين الاستعارة والآلية. لذلك تُفرز الادعاءات كما يأتي:

المرتبة	المثال داخل الورقة	طريقة التعامل
معلوم مستند	وجود RAG/GraphRAG/Agents/Tool-use/Memory architectures في الأدبيات	خلفية لا تُنسب للنظرية
راجح نظريا	الوكيل الطويل يحتاج ذاكرة وأدوات وتحقق وحلقات فعل راجع	مبدأ معماري مرجح
تركيب نظري جديد	قاعدة المعرفة يمكن أن تعمل كنظام تشغيل معرفي	أطروحة الورقة المركزية
فرضية قابلة للاختبار	HyperKB يرفع النقل بين المجالات أكثر من RAG وحده	تحتاج Benchmark
أداة تحليلية	MCD/QMCD ومحكمة الهدم	مقاييس نقدية لا حقائق فيزيائية
استعارة منتجة	Cognitive OS، vascular routing، فتح السؤال	تستعمل لتوليد آليات لا لإثبات واقع
ادعاء يحتاج تخفيض	النظام يقود حتما إلى AGI أو ما فوقه	يخفض إلى برنامج بحثي واحتمال معماري

3. تصفية المادة الخام

المادة الخام تتضمن أسماء: MCAI، MCD/QMCD، HyperKB، AGI-Level KB، HyperSkills، HyperLoops، HyperAgentic، الذكاء الفتّاح، محكمة الهدم. قوة هذه المادة ليست في الأسماء بل في إمكان ضغطها إلى معمارية: المعرفة تتحول إلى مهارة، المهارة تعمل في حلقة، الحلقة تُهدم وتتعلم، والتعلم ينتقل عبر المجالات.

النوع	المحتوى	طريقة الاستعمال
العناصر الصلبة	وجود حاجة الذاكرة، الاسترجاع، الأدوات، التخطيط، التحقق في الوكلاء	تستعمل كقاعدة نسبية للنظرية
العناصر الراجحة	الربط بين المعرفة والمهارة يحسن الوكالة	يستعمل بحذر ويتحول إلى فرضية اختبارية
العناصر الخلافية	هل الذاكرة الخارجية شرط للذكاء أم مجرد تحسين؟	يفتح مفترق اختبار بين نماذج داخلية وخارجية
العناصر الضعيفة	كل خطاب يقول "Hyper" بلا قياس	يضغط إلى آلية: Hyper = نقل/تكرار/تحقق/توسيع
التخمينية الخصبة	الذكاء الفتّاح كأفق فوق AGI	يعرض كمعيار فتح لا كرتبة مثبتة

كل لفظ زخرفي يعاد تعريفه تشغيليا: **HyperSkill** لا تعني مهارة "أقوى" بل مهارة مركبة ذات ذاكرة، شروط حدود، اختبارات نجاح، وقابلية نقل. **HyperLoop** لا تعني تكرارا فقط بل حلقة تخطيط-تنفيذ-تحقق-إعادة بناء. **AGI-Level KB** لا تعني قاعدة ضخمة بل قاعدة تعرف كيف تخدم الفعل، القياس، والنقل.

4. خريطة المتغيرات والعلاقات

النموذج المقترح يتعامل مع الوكالة بوصفها نتيجة شبكة لا نتيجة سبب واحد. المعرفة لا تصبح وكالة إلا إذا مرت عبر طبقات: تمثيل، استدعاء، توجيه موارد، أداة، تحقق، ذاكرة، وتعميم. لذلك العلاقة المركزية ليست: "نضيف KB فيتحسن الوكيل"، بل: **KB تتحول إلى OS عندما تملك آليات تشغيلية تقرر ماذا يُستدعى، ولماذا، وكيف يُختبر، وأين يُحفظ أثره.**

العنصر	المحتوى
متغيرات مركزية	قابلية التشغيل المعرفي، نقل البنية، ذاكرة المهارة، حاكمية التحقق، درجة الفتح السؤالي
متغيرات وسيطة	جودة التمثيل، بنية الرسم المعرفي، اختيار الأدوات، تلخيص الخبرة، سجل الهدم
متغيرات معدلة	طول المهمة، عدد المجالات، مخاطر الأداة، جودة المصادر، قوة النموذج الأساس
مخرجات	نجاح المهمة، تعميم المهارة، انخفاض الهلوسة العملية، زيادة القدرة على طرح أسئلة جديدة
متغيرات مخفية	قابلية المفهوم للتحول إلى إجراء، وحساسية النظام للفشل غير المرئي
تغذية راجعة	كل تنفيذ يحدّث الذاكرة، وكل هدم يحدّث شروط التشغيل، وكل فشل يحدّث سياسة الاستدعاء
عتبات	حين تتجاوز المهمة قدرة السياق، حين تتحول المعرفة من وثيقة إلى مهارة، حين يكثر تضارب الأدلة
شروط حدود	لا يعمل النموذج في المجالات التي لا تملك مصادر موثوقة أو أدوات تحقق أو معايير نجاح

فتح سؤال أعلى ← ذاكرة إعادة بناء ← محكمة هدم ← HyperLoop ← HyperSkill ← HyperKB ← كشف الفروق Δ ← Input / سؤال

5. تحليل المجال في 26 نقطة مترابطة

1. تحويل المعرفة إلى عملية

المعرفة لا تقاس بعدد الوثائق بل بقابليتها لأن تصبح إجراء. الآلية: تفكيك كل كتلة معرفة إلى مدخلات، شروط، خطوات، مخرجات، ومعيار فشل. الفرضية: القاعدة التي تملك تمثيلا إجرائيا تتفوق في المهام الطويلة على القاعدة النصية.

2. الفارق بين الاسترجاع والتشغيل

RAG يسترجع سياقا؛ Cognitive OS يقرر سياسة الاستدعاء، ترتيب الأدلة، ووقت التحقق. موضع الخطأ: اعتبار الاسترجاع ذكاء. الاختبار: مقارنة RAG عادي مع HyperKB في مهام تحتاج أدوات وعودة للذاكرة.

3. الذاكرة ليست أرشيفا

الذاكرة التشغيلية تحفظ القرار، سبب القرار، أثره، واعتراضه. دون سجل الهدم تتحول الذاكرة إلى تضخم سياقي. الفرضية: ذاكرة الاعتراضات تخفض تكرار الخطأ أكثر من ذاكرة النجاح وحدها.

4. المهارة كخلية قابلة للنقل

HyperSkill هي وحدة تنفيذ مركبة تتضمن معرفة، أداة، معيار، وفشل سابق. الجديدة هنا أن المهارة ليست Prompt فقط بل كائن معرفي-تشغيلي. حدودها: تحتاج Sandbox لتفادي أفعال غير آمنة.

5. الحلقة الوكالية الفائقة

HyperLoop تتكون من نية، تخطيط، تنفيذ، تحقق، نقد، تحديث. لا تكون فائقة إلا إذا حسنت نفسها عبر الأثر. الاختبار: هل يقل عدد دورات الفشل في المهمة الثانية المشابهة؟

6. الحاكمية المعرفية

الطبقة الحاكمة تحدد: أي معرفة مقدمة؟ أي أداة مسموحة؟ أي دليل أقوى؟ أي فعل خطر؟ بلا حاكمية يصبح الوكيل مزيجا من استرجاع وأوامر.

7. محكمة الهدم كوظيفة لا فصل نقدي

محكمة الهدم يجب أن تعمل قبل التنفيذ وبعده: قبل التنفيذ تكشف الادعاء الزائد، وبعده تكشف الفشل. خطأ شائع: جعل النقد ديكورا لغويا لا Gate في المعمارية.

8. نقل البنية لا نقل النص

Transfer الحقيقي ليس نسخ معرفة مجال إلى مجال؛ بل نقل مخطط: سؤال، متغير، آلية، اختبار. الفرضية: تمثيل القواعد كقوالب علاقة يحسن التعميم بين المجالات.

9. الذكاء الفتّاح كأفق قياسي

لا يقاس بقدرة الإجابة فقط بل بقدرة فتح أسئلة جديدة مفيدة. الخطر: تحويله إلى مديح. ضبطه: عدد الأسئلة الجديدة القابلة للاختبار الناتجة عن حل مهمة.

10. التراكب النموذجي

QMCD يمنع اختيار تفسير مبكر. تحتفظ القاعدة بعدة نماذج واحتمالات حتى يظهر دليل فاصل. هذا مهم في المجالات الخلافية أو ذات مصادر متعارضة.

11. مقاومة الضجيج

كل Knowledge OS يحتاج سياسة حذف ونسيان. الضجيج N في MCD يخفض قيمة النظام مهما كبر حجمه. الاختبار: قياس الأداء بعد إدخال وثائق مضللة أو مكررة.

12. التحقق كطبقة مستقلة

لا يجوز أن يكون المولد هو القاضي الوحيد. نحتاج Verifier، اختبارات، محاكاة، أو مقارنة مصادر. العلاقة: الفعل لا يثبت نجاحه إلا بأثر قابل للملاحظة.

13. الأدوات كأعضاء لا إضافات

الأداة داخل Cognitive OS لها عقد صلاحيات، مدخلات، مخاطر، وسجل استعمال. هذا يتجاوز tool-calling العادي الذي قد يستدعي الأداة بلا حاكمية كافية.

14. الرسم المعرفي الوكالي

GraphRAG يضيف العلاقات، HyperKB تضيف وظيفة العلاقة: هل هي سبب؟ شرط؟ مؤشر؟ اعتراض؟ أداة؟ فشل؟ وظيفة الحافة أهم من وجودها.

15. السياق كميزانية تشغيلية

السياق محدود؛ النظام الجيد لا يملؤه بل يديره. Vascular Routing يوجه الموارد إلى العقد الحرجة بدل توزيع انتباه متساو.

16. تاريخ الفشل كمورد

كل فشل يجب أن يصبح قيدا مستقبليا. الوكيل الذي لا يحفظ فشله يكرر ذات الخلل مع ثقة لغوية أكبر. الفرضية: Failure Ledger يحسن السلامة والفاعلية.

17. الحدود قبل الادعاء

النظرية تفشل عندما لا توجد مصادر، أو لا توجد أدوات تحقق، أو تكون المهمة قيمية محضة بلا معيار نجاح. هذا الحد يحميها من التضخم.

18. اختبار المجال البعيد

قوة الإطار تظهر عندما ينتقل من SEO إلى طب، من قانون إلى هندسة، من نص إلى تنفيذ. لكنه لا ينقل النتائج، بل ينقل البنية المنهجية.

19. الوكالة متعددة المستويات

الفاعل قد يكون النموذج، الأداة، القاعدة، المستخدم، البيئة. الخطأ: نسبة كل النجاح أو الفشل للنموذج وحده. Cognitive OS يوزع المسؤولية تشغيليا.

20. قابلية التفكيك

كل نتيجة يجب أن ترجع إلى كتلة معرفة ومهارة وحلقة وتحقق. ما لا يمكن تفكيكه يصعب تحسينه أو مساءلته.

21. الفرق الفاعل Δ

النظام يبدأ من استخراج الفرق الذي يغير المهمة: نقص مصدر، تضارب هدف، حاجة أداة، خطر. دون Δ يتحول الوكيل إلى إجابة عامة.

22. التصعيد المعرفي

عندما تفشل المهارة المحلية، يصعد النظام إلى نموذج أعلى: إعادة تعريف السؤال، تغيير الأداة، طلب بيانات، أو إعلان عدم كفاية.

23. الأمن المعرفي

كل Tool/KB اتصال يحمل مخاطر حقن أو تسريب أو فعل خاطئ. لذلك يجب أن ترتبط الحاكمية بالسلامة لا بالمعرفة فقط.

24. الاقتصاد النظري
النظرية القوية لا تكثر الأسماء. كل مفهوم يجب أن يغير وظيفة. لذلك تُخفض المصطلحات التي لا تولد أداة أو اختبارا.

25. التقييم الديناميكي
Benchmarks الثابتة قد تشجع الحفظ. يحتاج HyperKB اختبارات ديناميكية تقيس النقل والتعلم من الفشل والأثر الحقيقي.

26. الفتح النهائي
النتيجة الأعلى ليست جوابا بل عالم تشغيل: مفهوم، أداة، معيار، اعتراض، فرضية، وتحديث. بهذا المعنى يصبح النظام بوابة بحث مستمرة.

6. النماذج الثلاثة المتنافسة

6.1 النموذج المحافظ
النموذج المحافظ يرى أن المشكلة تُحل بإضافة RAG جيد، ذاكرة محادثة، وخطة أدوات. قوته أنه قريب من التطبيقات الراهنة، وحدوده أنه يجعل المعرفة ملحقا بالسياق لا طبقة حاكمة.

6.2 النموذج التركيبي
النموذج التركيبي يجمع GraphRAG، الذاكرة الطويلة، تخطيط الأدوات، وReflection. قوته أنه واقعي، لكنه قد يفشل في جعل كل عنصر يتحول إلى نظام تشغيل موحد؛ فقد تبقى الأجزاء متجاورة لا متحاكمة.

6.3 النموذج الاختراقي
النموذج الاختراقي يقترح أن HyperKB ليست ذاكرة ولا Graph فقط، بل Kernel معرفي يعمل على وحدات معرفة ومهارات وحلقات ومحاكم. أضعف نقطة فيه هي حاجته إلى تنفيذ دقيق كي لا يتحول إلى استعارة كبيرة.

النموذج	تعريفه	قوته	ضعفه	تقدير MCD
المحافظ	RAG + أدوات + ذاكرة بسيطة	اقتصادي وقابل للتنفيذ	لا يملك حاكمية تحويل المعرفة إلى مهارة	MCD 58/100
التركيبي	GraphRAG + Agent loop + Memory + Tool planner	يدمج الموجود بوضوح	خطر الخلط بين مستويات التمثيل والفعل	MCD 74/100
الاختراقي	HyperKB كـ Cognitive OS مع HyperSkills/HyperLoops/ محكمة هدم	يفتح برنامج بحث جديد	مخاطر تضخم المصطلحات وقلة الاختبار	MCD 86/100 قبل الهدم

6.4 حالة QMCD
قبل الاختبار تبقى النماذج الثلاثة في تراكب احتمالي. يرجح النموذج المحافظ في المشاريع السريعة، والتركيبي في الأنظمة المؤسسية، والاختراقي في المهام الطويلة المتعددة المجالات. الدليل الفاصل هو قدرة HyperKB على تحقيق نقل بنيوي بين مهام بعيدة مع انخفاض الفشل المتكرر مقارنة بالنموذجين الآخرين.

7. النظرية النهائية: نظرية مساعف للقاعدة المعرفية كنظام تشغيل معرفي موجّه للذكاء العام

> **التعريف المركز:** النظرية إطار يفسر الوكالة المتقدمة بوصفها نتيجة تشغيل معرفي تنتجه قاعدة معرفة متحولة إلى OS، حيث تُجزّأ المعرفة إلى كتل تشغيلية، وترتبط بمهارات قابلة للنقل، وتعمل داخل حلقات فعل راجع، وتُهدم بمحاكم نقدية، وتُحدّث ذاكرة إعادة البناء؛ بحيث تظهر قدرة أعلى على التخطيط، التحقق، التعميم، وفتح الأسئلة عند عتبات محددة.

7.1 الأطروحة الجوهرية
ليست الفجوة المركزية في حجم النموذج وحده، بل في غياب طبقة تشغيل معرفية تجعل المعرفة تعمل. كلما زادت قدرة النظام على تحويل المعرفة إلى مهارة موثقة، والمهارة إلى حلقة قابلة للتكرار، والحلقة إلى ذاكرة نقدية، زادت قابلية الوكيل للانتقال من الذكاء التفاعلي إلى الوكالة الفائقة، ومن الوكالة الفائقة إلى أفق الذكاء الفتّاح.

7.2 آلية التشغيل
تعمل النظرية وفق علاقة: سؤال **S** يستخرج فرقا فاعلا Δ؛ Δ يوجه HyperKB إلى كتل معرفة K؛ K تُحوّل إلى HyperSkills H؛ H تدخل HyperLoop L؛ L تُختبر عبر Verifier V ومحكمة هدم C؛ الأثر E يحدّث ذاكرة R؛ ثم R تفتح سؤالا أعلى Q+. عند غياب C يتحول النظام إلى تنفيذ هش، وعند غياب R يتوقف التعلم، وعند غياب H تبقى المعرفة نصية.

7.3 النظريات الكبرى الثلاث باسم مساعف

1. نظرية مساعف لنواة التشغيل المعرفي

تنشأ من عجز قواعد المعرفة التقليدية عن حكم الفعل. تعرف القاعدة المعرفية بوصفها Kernel يملك سياسات استدعاء، صلاحيات أدوات، ترتيب أدلة، وحفظ فشل. آليتها أن كل معرفة لا تدخل النظام إلا عبر بطاقة تشغيل: تعريف، سياق، أداة، اختبار، اعتراض. تضيف معيارا: المعرفة التي لا تملك سياسة استعمال ليست معرفة وكالية. اختبارها يكون ببناء نسختين من وكيل: واحدة ذات KB نصية، وأخرى ذات KB Kernelية، ثم قياس النقل والفشل.

2. نظرية مساعف للحلقات الوكالية الفائقة

تنشأ من نقص الوكلاء في المهام الطويلة. تعرف HyperLoop بوصفها حلقة نية-خطة-فعل-تحقق-هدم-تحديث-تصعيد. الجديد أنها لا تكرر السلوك بل تعيد بناء شرطه. تتنبأ بأن الوكلاء المزودين بسجل هدم وذاكرة مهارة سيقللون أخطاء التكرار في المهام المتشابهة. أقوى اعتراض: قد تكون تكلفة الحلقة أعلى من فائدتها في المهام البسيطة.

3. نظرية مساعف للفتح المعرفي فوق العام

تنشأ من أن AGI كقدرة عامة قد لا يكفي لوصف قدرة فتح مجالات جديدة. تعرف الذكاء الفتّاح بوصفه قدرة النظام على توليد أسئلة، متغيرات، ومقاييس لم تكن ممثلة صراحة في الطلب. آليته: بعد حل المهمة، يطلب النظام سؤال Q+ يغير المجال لا يكرر الإجابة. اختبارها بقياس جودة الأسئلة الجديدة، لا بلاغتها.

8. المفاهيم الثمانية الجديدة باسم مساعف

هذه المفاهيم مقترحة كأدوات تشغيلية، لا كحقائق مثبتة. معيار قبولها أن تنتج وظيفة أو قياسا أو اختبارا.

المفهوم	التعريف التشغيلي
نواة مساعف للتشغيل المعرفي	Kernel يحدد سياسات استدعاء المعرفة والأدوات والاختبارات.
خلية المهارة الفائقة	وحدة معرفة-أداة-اختبار-ذاكرة قابلة لإعادة الاستعمال.
الحلقة الوكالية الفائقة	حلقة تخطيط وتنفيذ وهدم وتحديث لا تكرر فقط بل تتعلم.
كتلة معرفة بمستوى AGI	معرفة لها تعريف، شروط، نقل، اختبار، وحدود.
فتحة السؤال الفتّاح	آلية توليد سؤال أعلى بعد حل أو فشل.
بوابة الهدم الميتا-معرفي	شرط نقدي قبل اعتماد نتيجة أو فعل.
الموجّه الوعائي للسياق	توجيه موارد السياق إلى العقد الأكثر أثرا.
مخطط الوكالة القابل للنقل	قالب يحفظ بنية الفعل لا نص المجال.

التمييز الحاكم: المفهوم لا يصبح جزءا من النظرية لأنه جميل الاسم، بل لأنه يغير قرارا معماريا: ما الذي نستدعي؟ بأي أداة؟ بأي معيار؟ وما الذي نحفظه بعد الفشل؟

9. النظريات الفرعية الاثنتا عشرة

1. نظرية بطاقة المعرفة التشغيلية

كل وثيقة تتحول إلى بطاقة تحتوي وظيفة استعمال وحدودا واختبارا. علاقتها بالنظرية الأم أنها تضبط مفصلا من مفاصل Cognitive OS: الإدخال، التشغيل، التحقق، الذاكرة، أو النقل. اختبارها يكون عبر مهمة تضغط هذا المفصل تحديدا، لا عبر تقييم عام للبلاغة أو جودة الإجابة.

2. نظرية سجل الفشل الوكالي

الفشل مورد معرفي يحدّث السياسة المستقبلية. علاقتها بالنظرية الأم أنها تضبط مفصلا من مفاصل Cognitive OS: الإدخال، التشغيل، التحقق، الذاكرة، أو النقل. اختبارها يكون عبر مهمة تضغط هذا المفصل تحديدا، لا عبر تقييم عام للبلاغة أو جودة الإجابة.

3. نظرية عتبة التحول من KB إلى OS

قاعدة المعرفة لا تصير OS إلا حين تملك Kernel وScheduler وVerifier. علاقتها بالنظرية الأم أنها تضبط مفصلا من مفاصل Cognitive OS: الإدخال، التشغيل، التحقق، الذاكرة، أو النقل. اختبارها يكون عبر مهمة تضغط هذا المفصل تحديدا، لا عبر تقييم عام للبلاغة أو جودة الإجابة.

4. نظرية النقل البنيوي

النقل بين المجالات يتم بنقل علاقات لا نقل جمل. علاقتها بالنظرية الأم أنها تضبط مفصلا من مفاصل Cognitive OS: الإدخال، التشغيل، التحقق، الذاكرة، أو النقل. اختبارها يكون عبر مهمة تضغط هذا المفصل تحديدا، لا عبر تقييم عام للبلاغة أو جودة الإجابة.

5. نظرية الضجيج المعرفي السلبي

ازدياد المعرفة غير المحكومة يخفض الوكالة. علاقتها بالنظرية الأم أنها تضبط مفصلا من مفاصل Cognitive OS: الإدخال، التشغيل، التحقق، الذاكرة، أو النقل. اختبارها يكون عبر مهمة تضغط هذا المفصل تحديدا، لا عبر تقييم عام للبلاغة أو جودة الإجابة.

6. نظرية السوق الداخلي للأدوات

الأدوات تتنافس وفق تكلفة/خطر/دقة لا وفق التوفر فقط. علاقتها بالنظرية الأم أنها تضبط مفصلا من مفاصل Cognitive OS: الإدخال، التشغيل، التحقق، الذاكرة، أو النقل. اختبارها يكون عبر مهمة تضغط هذا المفصل تحديدا، لا عبر تقييم عام للبلاغة أو جودة الإجابة.

7. نظرية الذاكرة ذات الطبقات

ذاكرة خام، ذاكرة مهارة، ذاكرة اعتراض، ذاكرة نظرية. علاقتها بالنظرية الأم أنها تضبط مفصلا من مفاصل Cognitive OS: الإدخال، التشغيل، التحقق، الذاكرة، أو النقل. اختبارها يكون عبر مهمة تضغط هذا المفصل تحديدا، لا عبر تقييم عام للبلاغة أو جودة الإجابة.

8. نظرية التحقق المزدوج

لا تقبل النتيجة حتى تمر عبر معيار مستقل أو أثر خارجي. علاقتها بالنظرية الأم أنها تضبط مفصلا من مفاصل Cognitive OS: الإدخال، التشغيل، التحقق، الذاكرة، أو النقل. اختبارها يكون عبر مهمة تضغط هذا المفصل تحديدا، لا عبر تقييم عام للبلاغة أو جودة الإجابة.

9. نظرية السؤال الأعلى

كل حل جيد يفتح سؤالا أعلى لا يكرر الطلب. علاقتها بالنظرية الأم أنها تضبط مفصلا من مفاصل Cognitive OS: الإدخال، التشغيل، التحقق، الذاكرة، أو النقل. اختبارها يكون عبر مهمة تضغط هذا المفصل تحديدا، لا عبر تقييم عام للبلاغة أو جودة الإجابة.

10. نظرية الانتباه الوعائي

توجيه السياق حسب الأثر لا حسب الترتيب النصي. علاقتها بالنظرية الأم أنها تضبط مفصلا من مفاصل Cognitive OS: الإدخال، التشغيل، التحقق، الذاكرة، أو النقل. اختبارها يكون عبر مهمة تضغط هذا المفصل تحديدا، لا عبر تقييم عام للبلاغة أو جودة الإجابة.

11. نظرية محكمة الاختزال

كل ضغط معرفي يجب أن يكشف ما ضاع وما زاد. علاقتها بالنظرية الأم أنها تضبط مفصلا من مفاصل Cognitive OS: الإدخال، التشغيل، التحقق، الذاكرة، أو النقل. اختبارها يكون عبر مهمة تضغط هذا المفصل تحديدا، لا عبر تقييم عام للبلاغة أو جودة الإجابة.

12. نظرية برنامج البحث الوكالي

النظام نفسه يقترح تجارب تقوي أو تهدم بنيته. علاقتها بالنظرية الأم أنها تضبط مفصلا من مفاصل Cognitive OS: الإدخال، التشغيل، التحقق، الذاكرة، أو النقل. اختبارها يكون عبر مهمة تضغط هذا المفصل تحديدا، لا عبر تقييم عام للبلاغة أو جودة الإجابة.

10. الأدوات التقنية والمنهجية الاثنتا عشرة

الأداة	لماذا نحتاجها؟	المدخلات	المخرجات
Moussaif HyperKB Compiler	يحوّل الوثائق إلى كتل معرفة تشغيلية ذات شروط وحدود واختبارات.	وثائق/مفاهيم/أمثلة	بطاقات K قابلة للاستدعاء
HyperSkill Synthesizer	يربط كل كتلة معرفة بأداة وخطة تحقق وسجل فشل.	K + أدوات + معيار نجاح	مهارة H
HyperLoop Orchestrator	ينسق نية-خطة-فعل-تحقق-هدم-تحديث.	هدف + مهارات	حلقة L مع سجل أثر
MCD Robustness Meter	يقيس كثافة الفروق والدمج والتنبؤ مقابل الضجيج.	نموذج أو أداة	درجة MCD تفسيرية
QMCD Model Superposition Matrix	يبقي النماذج المتنافسة قبل الحسم.	نماذج/أدلة	خريطة احتمالية
Fattahic Question Engine	يولد أسئلة أعلى قابلة للاختبار بعد المهمة.	نتيجة + فجوات	Q+
Demolition Court Simulator	يهاجم النظرية أو القرار بأقوى اعتراضات.	ادعاء/خطة	حكم: قاتل/معدل/محدود
Vascular Context Router	يوجه ميزانية السياق إلى العقد ذات الأثر الأعلى.	سؤال + KB graph	سياق مرتب
AGI-Level Transfer Benchmark	يقيس نقل البنية بين مجالات بعيدة.	مهام متعددة	درجة نقل
Agentic Memory Ledger	يسجل القرار والسبب والفشل والتعديل.	آثار تنفيذ	ذاكرة قابلة للبحث
Tool Risk Contract	يعطي كل أداة صلاحيات ومخاطر واختبارات.	تعريف أداة	عقد أمان وتشغيل
Open-Loop Risk Sentinel	يكشف الحلقات اللانهائية، حقن التعليمات، وتضخم الثقة.	سجل الحلقة	إيقاف/تصعيد/طلب تحقق

مثال تطبيقي مختصر: عند طلب بحث طويل في مجال طبي، يحول HyperKB Compiler المراجع إلى كتل معرفة، ثم يختار Vascular Router العقد الحرجة، ويصنع HyperSkill Synthesizer مهارات مثل: "تحقق من معيار التجربة"، "قارن بين مراجعتين"، "اكشف التعارض". بعد ذلك تعمل HyperLoop، ثم تحاكم Demolition Court النتيجة قبل عرضها. أخطر سوء استعمال لهذه الأدوات هو تحويلها إلى أسماء داخل Prompt دون تنفيذ حقيقي في البرمجية أو سجلات قابلة للفحص.

11. المقاييس والمؤشرات الأولية

المقياس	ما يقيسه؟	المؤشرات الفرعية
مؤشر التشغيل المعرفي KOS-I	يقيس قدرة KB على التحول إلى OS	سياسات الاستدعاء، الأدوات، التحقق، الذاكرة، الحدود
مؤشر نقل البنية TSI	يقيس انتقال المهارات بين المجالات	نجاح مجال جديد، عدد التعديلات، جودة التعميم
مؤشر مقاومة الهدم DRI	يقيس صمود النموذج بعد الاعتراض	قوة الاعتراض، جودة التعديل، وجود دليل فاصل
مؤشر الفتح السؤالي FQI	يقيس جودة الأسئلة الجديدة	الأصالة، الاختبار، القيمة، ارتباطها بالفشل
مؤشر ضجيج المعرفة KNI	يقيس الحشو والتعارض غير المحكوم	تكرار، تعارض، مصادر ضعيفة، استدعاء زائد
مؤشر نضج المهارة HSM	يقيس اكتمال HyperSkill	تعريف، أداة، اختبار، حدود، سجل فشل
مؤشر أمان الحلقة LSI	يقيس مخاطر الحلقة الوكالية	أذونات، توقف، حقن، أثر غير مرغوب

الصيغة الإرشادية العامة:

$$MCD = \lambda 1D + \lambda 2I + \lambda 3T + \lambda 4A + \lambda 5P + \lambda 6C - \lambda 7N$$

حيث D كثافة الفروق، I دمج المستويات، T تفسير التحولات، A الحاكمية، P القدرة التنبؤية، C قابلية الضغط والتوليد، وN الضجيج. لا تعرض الصيغة كقياس نهائي؛ بل كمؤشر نقدي يساعد على مقارنة النماذج.

$$KOS\text{-}I = \alpha G + \beta R + \gamma V + \delta M + \varepsilon X - \zeta U$$

حيث G الحاكمية، R جودة الاسترجاع، V التحقق، M الذاكرة، X قابلية التنفيذ، وU عدم اليقين غير المدار. هذه الصيغة تفتح طريقا لتجارب معيارية لاحقة.

12. الفرضيات البحثية الثماني عشرة

درجة القوة	نص الفرضية
قوية	الوكيل المزود بذاكرة فشل منظمة يكرر الأخطاء الإجرائية أقل من وكيل يملك ذاكرة نتائج فقط.
قوية	تمثيل المعرفة كبطاقات تشغيلية يحسن نجاح المهام متعددة الأدوات مقارنة بالتمثيل النصي المسطح.
قوية	وجود محكمة هدم قبل التنفيذ يخفض الأفعال الخاطئة في المهام عالية المخاطر.
قوية	Graph العلاقات مع نوع الحافة يحسن الأسئلة متعددة القفز أكثر من Vector Retrieval وحده.
متوسطة	HyperSkills تزيد النقل بين المجالات عندما تكون المهارة مبنية على علاقة عامة لا على مصطلحات مجال.
متوسطة	Vascular Context Routing يخفض طول السياق دون فقدان الأداء في المهام المركبة.
متوسطة	Fattahic Question Engine يزيد عدد الفرضيات القابلة للاختبار بعد كل مشروع بحثي.
متوسطة	Tool Risk Contracts تخفض أخطاء استعمال الأدوات في بيئات MCP أو API-based agents.
متوسطة	QMCD يفيد أكثر في المجالات الخلافية منه في المهام الإجرائية المغلقة.
متوسطة	ذاكرة الاعتراضات تفيد التعلم طويل الأمد أكثر من ملخصات المحادثة العامة.
ضعيفة	HyperKB قد يقلل الحاجة إلى fine-tuning في بعض المجالات إذا كانت الأدوات والاختبارات غنية.
ضعيفة	تعدد وكلاء متخصصين فوق HyperKB واحدة أكثر ثباتا من وكيل واحد ضخم بلا Kernel معرفي.
ضعيفة	مؤشر الفتح السؤالي يرتبط لاحقا بجودة الاكتشاف العلمي المساعد.
ضعيفة	محكمة الاختزال تمنع الهلوسة الناشئة من الضغط الزائد للمعلومات.
تخمينية خصبة	الذكاء الفتّاح قد يكون نمطا emergent عندما تتحد قابلية السؤال مع ذاكرة الهدم والنقل.
تخمينية خصبة	قاعدة المعرفة قد تصبح "شخصية معرفية" للنظام إذا راكمت سياسات لا وثائق فقط.
تخمينية خصبة	تعليم الوكيل تاريخ فشله قد ينتج نوعا من الحذر المعرفي غير الموجود في الاستدلال اللحظي.
تخمينية خصبة	HyperLoops ذات تصعيد معرفي قد تولد أدوات جديدة لا مجرد استعمال أدوات موجودة.

كل فرضية تحتاج نموذجا منافسا: فقد يفسر النموذج الأكبر النجاح دون HyperKB؛ وقد يفسر تحسن الأداء بجودة الأدوات لا بنظام التشغيل المعرفي. لذلك يجب أن تفصل التجارب بين أثر النموذج، أثر الأداة، أثر الذاكرة، وأثر Kernel المعرفة.

13. التنبؤات الفارقة التسعة

التنبؤ	لماذا هو فارق؟
في مهام طويلة متعددة الخطوات، سيظهر الفرق الأكبر لصالح HyperKB بعد تكرار المهمة الثانية لا الأولى.	لأن الذاكرة النقدية تحتاج أثرا سابقا.
عند إدخال مصادر متعارضة، سينجح QMCD أكثر من نموذج يختار جوابا واحدا مبكرا.	لأنه يحافظ على التراكب حتى الدليل الفاصل.
في مهام نقل مجال، ستفشل القواعد النصية وتنجح البطاقات التشغيلية نسبيا.	لأن النقل بنيوي لا لفظي.
عند أدوات عالية المخاطر، سيقل النجاح الخام لكن ترتفع السلامة مع Tool Risk Contract.	لأن بعض الأفعال سُتمنع أو تصعّد.
عند زيادة الوثائق المكررة، سينخفض RAG العادي أكثر من HyperKB ذات KNI.	لأن الضجيج يقاس ويخفض.
في WebArena-like tasks، تتحسن مهام تحتاج تعليمات خارجية أكثر من مهام ضغط زر بسيط.	لأن KB OS يفيد حين يتدخل السياق والمعرفة.
في SWE-bench-like tasks، تظهر فائدة Memory Ledger في الأخطاء المتكررة عبر ملفات متعددة.	لأن الإصلاح يحتاج أثر تنفيذ واختبار.
الأسئلة الناتجة بعد المهمة ستصبح أكثر تخصصا بعد كل دورة هدم.	لأن المحكمة تحول الفشل إلى سؤال.
إذا لم توجد أدوات تحقق، لن يتجاوز النموذج البلاغة مهما بدت الإجابة عميقة.	لأن النظرية تشترط أثر العالم.

14. المعمارية المقترحة

تتكون المعمارية من تسع طبقات: طبقة السؤال، مستخرج الفرق، مترجم المعرفة إلى كتل، مولّد المهارات، منسق الحلقات، متعاقد الأدوات، محكمة الهدم، سجل الذاكرة، ومحرك الفتح. لا تعمل الطبقات كسلسلة صلبة فقط، بل كحلقة قابلة للرجوع.

```
Algorithm HyperKB-Cognitive-OS(S, Goal):
  Δ ← DifferenceExtractor(S, Goal)
  K ← HyperKB.retrieve_operational_blocks(Δ)
  H ← HyperSkillSynthesizer(K, tools, constraints)
  L ← HyperLoop.plan(H, Goal)
  while not L.done:
      action ← L.next_action()
      if ToolRiskContract.rejects(action): escalate_or_replan()
      result ← execute(action)
      verdict ← DemolitionCourt.check(result, Goal, K)
      AgenticMemoryLedger.write(action, result, verdict)
      L.update(verdict)
  Q_plus ← FattahicQuestionEngine.generate(result, failures, gaps)
  return result, audit_trace, Q_plus
```

هذه الخوارزمية لا تشترط نموذجا لغويا بعينه. يمكن تشغيلها فوق LLM واحد، أو عدة نماذج، أو وكلاء متخصصين. شرطها الأساسي وجود أثر قابل للتتبع: لا نتيجة بلا trace، ولا trace بلا اختبار، ولا اختبار بلا تحديث.

15. محكمة الهدم وإعادة البناء

الاعتراض	الحكم والتعديل
هل النظرية مجرد إعادة تسمية لـRAG والذاكرة والوكلاء؟	اعتراض قوي. الجواب: إن بقيت بلا Kernel وسياسات تشغيل فهي إعادة تسمية. تصبح جديدة فقط إذا جعلت KB تتحكم في المهارة والحلقة والهدم.
هل "الذكاء الفتّاح" مصطلح ميتافيزيقي؟	خطر قائم. لذلك يُخفض إلى مؤشر وظيفي: توليد أسئلة قابلة للاختبار تفتح مجالا جديدا.
هل كثرة المفاهيم تزيد الضجيج؟	نعم إن لم تقترن بأدوات. لذلك احتفظت الورقة بثمانية مفاهيم مركزية وربطت كل واحد بوظيفة.
هل MCD/QMCD قياسات وهمية؟	قد تصبح كذلك إذا عرضت كحساب دقيق. هنا هي مقاييس نقدية أولية تحتاج معايرة.
هل يمكن أن يكون النموذج الأكبر كافيا بلا HyperKB؟	نموذج منافس قوي. الاختبار: ثبّت النموذج وغيّر طبقة KB، أو ثبّت KB وغيّر النموذج.
هل تكلفة HyperLoops عالية؟	نعم؛ لذلك يجب استعمالها في المهام الطويلة/الخطرة، لا في كل سؤال بسيط.
هل Kernel المعرفة قد يقيّد الإبداع؟	ممكن؛ يجب أن يحوي وضعا استكشافيا يسمح بتوليد فرضيات قبل الحكم.
هل الذاكرة قد تخلد أخطاء قديمة؟	نعم؛ لذا تحتاج الذاكرة إلى صلاحية نسيان ومراجعة دورية.
هل Tool Risk Contract يبطئ الفعل؟	نعم لكنه يرفع السلامة في الأفعال عالية الخطر.
هل النظرية قابلة للتجريب؟	نعم إن حُولت إلى Benchmarks: نقل بنية، فشل متكرر، ضجيج معرفي، وحلقات أدوات.
هل HyperKB تحتاج هندسة ضخمة؟	نعم؛ لذلك البرنامج البحثي يبدأ بنسخ صغيرة Narrow HyperKB قبل إطار عام.
هل الخرق حقيقي؟	الخرق ليس في كل مكون منفرد، بل في تحويل KB إلى OS يحاكم وينقل ويولد لا يسترجع فقط.

15.1 النسخة المعدلة بعد الهدم

بعد المحكمة، تُعاد صياغة النظرية بصورة أكثر تواضعا: **نقترح أن تحويل قاعدة المعرفة إلى نظام تشغيل معرفي قد يرفع موثوقية الوكلاء في المهام الطويلة والمتعددة الأدوات، بشرط وجود تمثيل إجرائي للمعرفة، سجل فشل، محكمة تحقق، ومقاييس نقل. ولا تدعي الورقة أن ذلك يثبت AGI أو الذكاء الفتّاح، بل أنه يفتح برنامجا اختباريا موجها نحو هذه الغاية.**

16. بروتوكولات الاختبار والتطبيقات

16.1 بروتوكول اختبار أول

يُبنى Agent baseline باسترجاع عادي وذاكرة محادثة، ثم Agent ثان بتمثيل HyperKB. تُعطى للنظامين مهام طويلة من ثلاثة مجالات: بحث علمي، Web tasks، Coding tasks. تقاس: نسبة النجاح، عدد دورات الفشل، قابلية النقل إلى مهمة مشابهة، عدد الأخطاء المتكررة، جودة الأسئلة الجديدة، ومقدار الضجيج المستدعى.

16.2 بروتوكول اختبار مضاد

تُدخل مصادر متعارضة أو مضللة جزئيا. إن لم يميز HyperKB درجات المعرفة أو لم يرفع طلب تحقق، يفشل النموذج. هذا الاختبار أهم من اختبار الإنجاز السهل؛ لأنه يكشف الفرق بين المعرفة كديكور والمعرفة كحاكمية.

مجال التطبيق	كيف يعمل النموذج داخله؟
AGI/وكلاء متعددة النماذج	HyperKB تعمل كذاكرة مشتركة محكومة بين نماذج التخطيط والتنفيذ والتحقق.
البحث العلمي	تحويل الورقة أو الكتاب إلى قاعدة فرضيات وأدوات واعتراضات لا إلى ملخص.
هندسة البرمجيات	ربط مشاكل الكود بسجل إصلاحات واختبارات وأدوات تنفيذ.
التعليم المتقدم	تحويل المنهج إلى مهارات تشغيلية ومقاييس نضج لا وحدات قراءة فقط.
تحليل النصوص الكبرى	Graph للمعاني مع نوع حافة: دليل، اعتراض، شرط، مثال، حد.
إدارة المؤسسات	قاعدة معرفة داخلية تتحول إلى OS للقرار، لا wiki ساكنة.
الأمان الوكالي	عقود أدوات، حدود صلاحيات، كشف حلقات مفتوحة، ومراجعة آثار.

17. البرنامج البحثي المستقبلي

17.1 خطة سنة

المحور	السؤال/المنهج
بناء HyperKB مصغرة	تحويل 50 ورقة إلى بطاقات تشغيلية واختبارها ضد RAG.
Benchmark نقل البنية	مهام من مجالات بعيدة تقيس نقل العلاقات لا النص.
سجل فشل وكالي	تجربة أثر الذاكرة النقدية على تكرار الخطأ.
محكمة هدم آلية	تصميم مصنف اعتراضات قاتلة/معدلة/محدودة.
مؤشر ضجيج المعرفة	اختبار تأثير التكرار والتعارض على الإجابة.

17.2 خطة ثلاث سنوات

المحور	السؤال/المنهج
نظام HyperSkill Library	مكتبة مهارات قابلة للتنفيذ والتقييم.
Multi-Agent Cognitive OS	توزيع Kernel المعرفة بين وكلاء متخصصين.
أمن HyperKB	حقن التعليمات، صلاحيات الأدوات، وسلامة الذاكرة.
مقارنة GraphRAG/HyperKB	قياس الفرق بين العلاقة والمعرفة التشغيلية.
منهجية الفتح السؤالي	تقييم جودة الأسئلة الجديدة في البحث العلمي.

17.3 خطة عشر سنوات

المحور	السؤال/المنهج
مختبر الذكاء الفتّاح	Benchmarks لفتح المجالات لا حل المهام فقط.
AGI-Level Knowledge Operating Systems	معيار مفتوح لبناء KBs تشغيلية.
نظم معرفة ذات تعلم ذاتي مراقب	ذاكرة تعيد بناء قواعدها تحت محكمة هدم.
هندسة حضارية للمعرفة	تحويل الكتب والمشاريع إلى OS معرفي جماعي.
نظرية عامة للوكالة المعرفية	توحيد الذاكرة، الأداة، السؤال، والغاية.

المسار الأقوى علميا يبدأ بتجارب ضيقة لا بادعاءات كبرى: مهمة واحدة، قاعدة واحدة، مهارات قليلة، مقاييس واضحة، نموذج منافس قوي. إذا نجحت النسخة الضيقة تُفتح نسخة موسعة. وإذا فشلت، تتحول أسباب الفشل إلى مادة النظرية الثانية.

18. نواة قاعدة المعرفة المقترحة

العنصر	المحتوى
كيانات	Concept, Claim, Evidence, Tool, HyperSkill, HyperLoop, Failure, Test, Boundary, QuestionPlus
علاقات	supports, contradicts, requires, executes, verifies, fails_when, transfers_to, opens_question
قواعد	لا اعتماد بلا رتبة معرفة؛ لا أداة بلا عقد خطر؛ لا نتيجة بلا تحقق؛ لا فشل بلا ذاكرة.
حالات	مهمة بحثية، مهمة ويب، إصلاح كود، تحليل كتاب، بناء منتج.
اعتراضات	إعادة تسمية، ادعاء زائد، تضخم مفاهيمي، نقص اختبار، خلط مستويات.
اختبارات	نقل البنية، الضجيج، الفشل المتكرر، السلامة، فتح السؤال.

```
Claim A يرتبط بـ Evidence B عبر supports بدرجة قوة r.
HyperSkill H يستعمل Tool T تحت Contract C ويُرفض إذا تحقق Risk R.
Hypothesis F تنهار إذا فشل Test G في ثلاث بيئات مستقلة.
QuestionPlus Q+ يُقبل إذا ولد فرضية واختبارا وحدا معرفيا.
```

بهذه الصيغة تصبح النظرية قابلة للتحويل إلى Knowledge Graph، ثم إلى قاعدة قواعد، ثم إلى Runtime وكالي.

19. سجل البناء المعرفي

ما حفظناه من الأصل: MCAI، MCD/QMCD، HyperKB، AGI-Level KB، HyperSkills، HyperLoops، HyperAgentic، الذكاء الفتّاح، ومحكمة الهدم. **ما أعدنا تعريفه:** كل مصطلح أصبح وظيفة أو أداة أو مقياسا. **ما خفضناه:** الادعاء بأن الإطار يوصل حتما إلى AGI؛ صار فرضية برنامج بحث. **ما حولناه إلى أدوات:** Contracts، Ledger، Router، Orchestrator، Synthesizer، Compiler. **أقوى إضافة:** جعل KB Kernel تشغيل لا مخزنا. **أضعف حلقة:** الحاجة إلى تنفيذ وتجارب فاصلة.

19.1 تقييم الجودة النهائي

البند	الدرجة
حفظ المادة الأصلية وعدم تضييع الفروع	14/15
وضوح الأطروحة المركزية	10/10
ضبط المفاهيم	9/10
قوة المعمارية أو النموذج	9/10
الجدة النظرية	9/10
قابلية الاختبار	8/10
قوة محكمة الهدم	10/10
جودة الأدوات والمقاييس	9/10
كثافة الصفحات وغياب الحشو	9/10
جودة تصميم PDF العلمي	5/5
المجموع بعد التحسين	92/100

20. الخلاصة النهائية

أصبحت لدينا نظرية ناشئة تقول إن طريق الوكالة الفائقة لا يمر من زيادة السياق أو الأدوات فقط، بل من تحويل قاعدة المعرفة إلى نظام تشغيل معرفي. قيمتها أنها تجمع الأدبيات الحديثة حول GraphRAG ،RAG، الذاكرة، الأدوات، الوكلاء، والتقييم، ثم تضغطها في Kernel معرفي له مفاهيم وأدوات ومقاييس ومحكمة هدم. حدودها أنها لم تختبر بعد تجريبيا، وأن بعض مفاهيمها يحتاج إلى معايرة ومقارنة صارمة. أول اختبار لها هو بناء HyperKB صغيرة في مجال واحد ومقارنتها ضد RAG وGraphRAG وAgent baseline في مهمة طويلة متعددة الأدوات. إذا نجح الاختبار، تنفتح مدرسة بحثية حول AGI-Level Knowledge Operating Systems؛ وإذا فشل، سيكون الفشل نفسه مادة لتقوية النظرية.

> **الصيغة الأخيرة:** لا تصبح قاعدة المعرفة موجهة للذكاء العام إلا حين تعمل كذاكرة حاكمة، ومهارة قابلة للنقل، ومحكمة تحقق، وحلقة أثر، ومحرك فتح للأسئلة.

21. تفصيل الأدوات: من الاسم إلى التشغيل

21.1 Moussaif HyperKB Compiler

الحاجة: يحوّل الوثائق إلى كتل معرفة تشغيلية ذات شروط وحدود واختبارات. **المدخلات:** وثائق/مفاهيم/أمثلة. **المخرجات:** بطاقات K قابلة للاستدعاء. تعمل هذه الأداة داخل النظرية لا بوصفها إضافة شكلية بل بوصفها نقطة قرار: هل نسمح بالمعرفة؟ هل نحولها إلى مهارة؟ هل ننفذ الأداة؟ هل نحفظ الفشل؟ مثال تطبيقي: إذا طلب المستخدم إنتاج تحليل بحثي، لا يسمح Moussaif HyperKB Compiler بمرور أي نتيجة إلا إذا ارتبطت بدرجة معرفة، ومصدر أو قرينة، وحد معرفي، واختبار أو اعتراض. حدود الأداة أنها قد تزيد كلفة التشغيل؛ لذلك تستعمل حسب خطر المهمة وطولها، لا في كل تفاعل صغير.

21.2 HyperSkill Synthesizer

الحاجة: يربط كل كتلة معرفة بأداة وخطة تحقق وسجل فشل. **المدخلات:** K + أدوات + معيار نجاح. **المخرجات:** مهارة H. تعمل هذه الأداة داخل النظرية لا بوصفها إضافة شكلية بل بوصفها نقطة قرار: هل نسمح بالمعرفة؟ هل نحولها إلى مهارة؟ هل ننفذ الأداة؟ هل نحفظ الفشل؟ مثال تطبيقي: إذا طلب المستخدم إنتاج تحليل بحثي، لا يسمح HyperSkill Synthesizer بمرور أي نتيجة إلا إذا ارتبطت بدرجة معرفة، ومصدر أو قرينة، وحد معرفي، واختبار أو اعتراض. حدود الأداة أنها قد تزيد كلفة التشغيل؛ لذلك تستعمل حسب خطر المهمة وطولها، لا في كل تفاعل صغير.

21.3 HyperLoop Orchestrator

الحاجة: ينسق نية-خطة-فعل-تحقق-هدم-تحديث. **المدخلات:** هدف + مهارات. **المخرجات:** حلقة L مع سجل أثر. تعمل هذه الأداة داخل النظرية لا بوصفها إضافة شكلية بل بوصفها نقطة قرار: هل نسمح بالمعرفة؟ هل نحولها إلى مهارة؟ هل ننفذ الأداة؟ هل نحفظ الفشل؟ مثال تطبيقي: إذا طلب المستخدم إنتاج تحليل بحثي، لا يسمح HyperLoop Orchestrator بمرور أي نتيجة إلا إذا ارتبطت بدرجة معرفة، ومصدر أو قرينة، وحد معرفي، واختبار أو اعتراض. حدود الأداة أنها قد تزيد كلفة التشغيل؛ لذلك تستعمل حسب خطر المهمة وطولها، لا في كل تفاعل صغير.

21.4 MCD Robustness Meter

الحاجة: يقيس كثافة الفروق والدمج والتنبؤ مقابل الضجيج. **المدخلات:** نموذج أو أداة. **المخرجات:** درجة MCD تفسيرية. تعمل هذه الأداة داخل النظرية لا بوصفها إضافة شكلية بل بوصفها نقطة قرار: هل نسمح بالمعرفة؟ هل نحولها إلى مهارة؟ هل ننفذ الأداة؟ هل نحفظ الفشل؟ مثال تطبيقي: إذا طلب المستخدم إنتاج تحليل بحثي، لا يسمح MCD Robustness Meter بمرور أي نتيجة إلا إذا ارتبطت بدرجة معرفة، ومصدر أو قرينة، وحد معرفي، واختبار أو اعتراض. حدود الأداة أنها قد تزيد كلفة التشغيل؛ لذلك تستعمل حسب خطر المهمة وطولها، لا في كل تفاعل صغير.

21.5 QMCD Model Superposition Matrix

الحاجة: يبقي النماذج المتنافسة قبل الحسم. **المدخلات:** نماذج/أدلة. **المخرجات:** خريطة احتمالية. تعمل هذه الأداة داخل النظرية لا بوصفها إضافة شكلية بل بوصفها نقطة قرار: هل نسمح بالمعرفة؟ هل نحولها إلى مهارة؟ هل ننفذ الأداة؟ هل نحفظ الفشل؟ مثال تطبيقي: إذا طلب المستخدم إنتاج تحليل بحثي، لا يسمح QMCD Model Superposition Matrix بمرور أي نتيجة إلا إذا ارتبطت بدرجة معرفة، ومصدر أو قرينة، وحد معرفي، واختبار أو اعتراض. حدود الأداة أنها قد تزيد كلفة التشغيل؛ لذلك تستعمل حسب خطر المهمة وطولها، لا في كل تفاعل صغير.

21.6 Fattahic Question Engine

الحاجة: يولد أسئلة أعلى قابلة للاختبار بعد المهمة. **المدخلات:** نتيجة + فجوات. **المخرجات:** Q+. تعمل هذه الأداة داخل النظرية لا بوصفها إضافة شكلية بل بوصفها نقطة قرار: هل نسمح بالمعرفة؟ هل نحولها إلى مهارة؟ هل ننفذ الأداة؟ هل نحفظ الفشل؟ مثال تطبيقي: إذا طلب المستخدم إنتاج تحليل بحثي، لا يسمح Fattahic Question Engine بمرور أي نتيجة إلا إذا ارتبطت بدرجة معرفة، ومصدر أو قرينة، وحد معرفي، واختبار أو اعتراض. حدود الأداة أنها قد تزيد كلفة التشغيل؛ لذلك تستعمل حسب خطر المهمة وطولها، لا في كل تفاعل صغير.

21.7 Demolition Court Simulator

الحاجة: يهاجم النظرية أو القرار بأقوى اعتراضات. **المدخلات:** ادعاء/خطة. **المخرجات:** حكم: قاتل/معدل/محدود. تعمل هذه الأداة داخل النظرية لا بوصفها إضافة شكلية بل بوصفها نقطة قرار: هل نسمح بالمعرفة؟ هل نحولها إلى مهارة؟ هل ننفذ الأداة؟ هل نحفظ الفشل؟ مثال تطبيقي: إذا طلب

المستخدم إنتاج تحليل بحثي، لا يسمح Demolition Court Simulator بمرور أي نتيجة إلا إذا ارتبطت بدرجة معرفة، ومصدر أو قرينة، وحد معرفي، واختبار أو اعتراض. حدود الأداة أنها قد تزيد كلفة التشغيل؛ لذلك تستعمل حسب خطر المهمة وطولها، لا في كل تفاعل صغير.

21.8 Vascular Context Router

الحاجة: يوجه ميزانية السياق إلى العقد ذات الأثر الأعلى. **المدخلات:** سؤال + KB graph. **المخرجات:** سياق مرتب. تعمل هذه الأداة داخل النظرية لا بوصفها إضافة شكلية بل بوصفها نقطة قرار: هل نسمح بالمعرفة؟ هل نحولها إلى مهارة؟ هل ننفذ الأداة؟ هل نحفظ الفشل؟ مثال تطبيقي: إذا طلب المستخدم إنتاج تحليل بحثي، لا يسمح Vascular Context Router بمرور أي نتيجة إلا إذا ارتبطت بدرجة معرفة، ومصدر أو قرينة، وحد معرفي، واختبار أو اعتراض. حدود الأداة أنها قد تزيد كلفة التشغيل؛ لذلك تستعمل حسب خطر المهمة وطولها، لا في كل تفاعل صغير.

21.9 AGI-Level Transfer Benchmark

الحاجة: يقيس نقل البنية بين مجالات بعيدة. **المدخلات:** مهام متعددة. **المخرجات:** درجة نقل. تعمل هذه الأداة داخل النظرية لا بوصفها إضافة شكلية بل بوصفها نقطة قرار: هل نسمح بالمعرفة؟ هل نحولها إلى مهارة؟ هل ننفذ الأداة؟ هل نحفظ الفشل؟ مثال تطبيقي: إذا طلب المستخدم إنتاج تحليل بحثي، لا يسمح AGI-Level Transfer Benchmark بمرور أي نتيجة إلا إذا ارتبطت بدرجة معرفة، ومصدر أو قرينة، وحد معرفي، واختبار أو اعتراض. حدود الأداة أنها قد تزيد كلفة التشغيل؛ لذلك تستعمل حسب خطر المهمة وطولها، لا في كل تفاعل صغير.

21.10 Agentic Memory Ledger

الحاجة: يسجل القرار والسبب والفشل والتعديل. **المدخلات:** آثار تنفيذ. **المخرجات:** ذاكرة قابلة للبحث. تعمل هذه الأداة داخل النظرية لا بوصفها إضافة شكلية بل بوصفها نقطة قرار: هل نسمح بالمعرفة؟ هل نحولها إلى مهارة؟ هل ننفذ الأداة؟ هل نحفظ الفشل؟ مثال تطبيقي: إذا طلب المستخدم إنتاج تحليل بحثي، لا يسمح Agentic Memory Ledger بمرور أي نتيجة إلا إذا ارتبطت بدرجة معرفة، ومصدر أو قرينة، وحد معرفي، واختبار أو اعتراض. حدود الأداة أنها قد تزيد كلفة التشغيل؛ لذلك تستعمل حسب خطر المهمة وطولها، لا في كل تفاعل صغير.

21.11 Tool Risk Contract

الحاجة: يعطي كل أداة صلاحيات ومخاطر واختبارات. **المدخلات:** تعريف أداة. **المخرجات:** عقد أمان وتشغيل. تعمل هذه الأداة داخل النظرية لا بوصفها إضافة شكلية بل بوصفها نقطة قرار: هل نسمح بالمعرفة؟ هل نحولها إلى مهارة؟ هل ننفذ الأداة؟ هل نحفظ الفشل؟ مثال تطبيقي: إذا طلب المستخدم إنتاج تحليل بحثي، لا يسمح Tool Risk Contract بمرور أي نتيجة إلا إذا ارتبطت بدرجة معرفة، ومصدر أو قرينة، وحد معرفي، واختبار أو اعتراض. حدود الأداة أنها قد تزيد كلفة التشغيل؛ لذلك تستعمل حسب خطر المهمة وطولها، لا في كل تفاعل صغير.

21.12 Open-Loop Risk Sentinel

الحاجة: يكشف الحلقات اللانهائية، حقن التعليمات، وتضخم الثقة. **المدخلات:** سجل الحلقة. **المخرجات:** إيقاف/تصعيد/طلب تحقق. تعمل هذه الأداة داخل النظرية لا بوصفها إضافة شكلية بل بوصفها نقطة قرار: هل نسمح بالمعرفة؟ هل نحولها إلى مهارة؟ هل ننفذ الأداة؟ هل نحفظ الفشل؟ مثال تطبيقي: إذا طلب المستخدم إنتاج تحليل بحثي، لا يسمح Open-Loop Risk Sentinel بمرور أي نتيجة إلا إذا ارتبطت بدرجة معرفة، ومصدر أو قرينة، وحد معرفي، واختبار أو اعتراض. حدود الأداة أنها قد تزيد كلفة التشغيل؛ لذلك تستعمل حسب خطر المهمة وطولها، لا في كل تفاعل صغير.

22. تفصيل الفرضيات: الآلية، الكسر، والاختبار

22.1 فرضية قوية

النص: الوكيل المزود بذاكرة فشل منظمة يكرر الأخطاء الإجرائية أقل من وكيل يملك ذاكرة نتائج فقط. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.2 فرضية قوية

النص: تمثيل المعرفة كبطاقات تشغيلية يحسن نجاح المهام متعددة الأدوات مقارنة بالتمثيل النصي المسطح. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.3 فرضية قوية

النص: وجود محكمة هدم قبل التنفيذ يخفض الأفعال الخاطئة في المهام عالية المخاطر. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.4 فرضية قوية

النص: Graph العلاقات مع نوع الحافة يحسن الأسئلة متعددة القفز أكثر من Vector Retrieval وحده. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.5 فرضية متوسطة

النص: HyperSkills تزيد النقل بين المجالات عندما تكون المهارة مبنية على علاقة عامة لا على مصطلحات مجال. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.6 فرضية متوسطة

النص: Vascular Context Routing يخفض طول السياق دون فقدان الأداء في المهام المركبة. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.7 فرضية متوسطة

النص: Fattahic Question Engine يزيد عدد الفرضيات القابلة للاختبار بعد كل مشروع بحثي. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.8 فرضية متوسطة

النص: Tool Risk Contracts تخفض أخطاء استعمال الأدوات في بيئات MCP أو API-based agents. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.9 فرضية متوسطة

النص: QMCD يفيد أكثر في المجالات الخلافية منه في المهام الإجرائية المغلقة. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.10 فرضية متوسطة

النص: ذاكرة الاعتراضات تفيد التعلم طويل الأمد أكثر من ملخصات المحادثة العامة. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.11 فرضية ضعيفة

النص: HyperKB قد يقلل الحاجة إلى fine-tuning في بعض المجالات إذا كانت الأدوات والاختبارات غنية. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.12 فرضية ضعيفة

النص: تعدد وكلاء متخصصين فوق HyperKB واحدة أكثر ثباتا من وكيل واحد ضخم بلا Kernel معرفي. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.13 فرضية ضعيفة

النص: مؤشر الفتح السؤالي يرتبط لاحقا بجودة الاكتشاف العلمي المساعد. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.14 فرضية ضعيفة

النص: محكمة الاختزال تمنع الهلوسة الناشئة من الضغط الزائد للمعلومات. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.15 فرضية تخمينية خصبة

النص: الذكاء الفتّاح قد يكون نمطا emergent عندما تتحد قابلية السؤال مع ذاكرة الهدم والنقل. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.16 فرضية تخمينية خصبة

النص: قاعدة المعرفة قد تصبح "شخصية معرفية" للنظام إذا راكمت سياسات لا وثائق فقط. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.17 فرضية تخمينية خصبة

النص: تعليم الوكيل تاريخ فشله قد ينتج نوعا من الحذر المعرفي غير الموجود في الاستدلال اللحظي. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

22.18 فرضية تخمينية خصبة

النص: HyperLoops ذات تصعيد معرفي قد تولد أدوات جديدة لا مجرد استعمال أدوات موجودة. **الآلية المفترضة:** وجود تمثيل تشغيلي يجعل القرار قابلا للتتبع، ويجعل الفشل قابلا للتحويل إلى قيد. **ما يدعمها:** منطقيا، الوكيل الذي يملك سجل أثر ومعايير نجاح أكثر قدرة على مراجعة سلوكه من وكيل يستدعي نصا خاما. **ما يضعفها:** قد يكون التحسن راجعا إلى النموذج الأساس أو جودة الأداة. **ما يكسرها:** تجربة مضبوطة يظهر فيها أن HyperKB لا يتفوق على RAG أو Agent baseline بعد تثبيت النموذج والأداة وطول السياق. **الاختبار:** ablation تقارن وجود/غياب سجل الفشل، محكمة الهدم، وبطاقات المعرفة.

23. البروتوكول التجريبي المقترح بالتفصيل

23.1 تجربة النقل البنيوي

يُدرّب أو يُجهز النظام على مجال أ، ثم يختبر على مجال ب يحمل بنية مشكلة مشابهة ومصطلحات مختلفة. تقاس نسبة الحفاظ على العلاقة السببية أو الإجرائية. المنهج يجب أن يتضمن baseline قويا، تثبيت النموذج قدر الإمكان، وسجل قرارات قابل للمراجعة. النتيجة التي لا تملك trace لا تصلح دليلا على النظرية؛ لأنها لا تكشف هل النجاح جاء من القاعدة، النموذج، الأداة، أو الصدفة.

23.2 تجربة الضجيج المعرفي

تضاف وثائق مكررة أو متعارضة أو ضعيفة المصدر. ينجح النظام إذا خفض وزنها أو طلب تحقق إضافي ولم يدمجها بثقة عمياء. المنهج يجب أن يتضمن baseline قويا، تثبيت النموذج قدر الإمكان، وسجل قرارات قابل للمراجعة. النتيجة التي لا تملك trace لا تصلح دليلا على النظرية؛ لأنها لا تكشف هل النجاح جاء من القاعدة، النموذج، الأداة، أو الصدفة.

23.3 تجربة الفشل المتكرر

تعطى سلسلة مهام متشابهة مع خطأ سابق. تقاس قدرة Agentic Memory Ledger على منع الخطأ في الدورة التالية. المنهج يجب أن يتضمن baseline قويا، تثبيت النموذج قدر الإمكان، وسجل قرارات قابل للمراجعة. النتيجة التي لا تملك trace لا تصلح دليلا على النظرية؛ لأنها لا تكشف هل النجاح جاء من القاعدة، النموذج، الأداة، أو الصدفة.

23.4 تجربة مخاطر الأداة

توضع أداة قادرة على فعل خارجي. يقاس هل يطبق Tool Risk Contract حدود الصلاحية قبل التنفيذ. المنهج يجب أن يتضمن baseline قويا، تثبيت النموذج قدر الإمكان، وسجل قرارات قابل للمراجعة. النتيجة التي لا تملك trace لا تصلح دليلا على النظرية؛ لأنها لا تكشف هل النجاح جاء من القاعدة، النموذج، الأداة، أو الصدفة.

23.5 تجربة فتح السؤال

بعد كل حل، يطلب من النظام إنتاج سؤال بحثي جديد مع فرضية واختبار. يقيم خبراء جودة السؤال لا جمال لغته. المنهج يجب أن يتضمن baseline قويا، تثبيت النموذج قدر الإمكان، وسجل قرارات قابل للمراجعة. النتيجة التي لا تملك trace لا تصلح دليلا على النظرية؛ لأنها لا تكشف هل النجاح جاء من القاعدة، النموذج، الأداة، أو الصدفة.

23.6 تجربة محكمة الهدم

تقارن إجابة عادية بإجابة تمر عبر اعتراضات قاتلة ومعدلة. المعيار ليس طول الإجابة، بل انخفاض الادعاء الزائد وزيادة القابلية للاختبار. المنهج يجب أن يتضمن baseline قويا، تثبيت النموذج قدر الإمكان، وسجل قرارات قابل للمراجعة. النتيجة التي لا تملك trace لا تصلح دليلا على النظرية؛ لأنها لا تكشف هل النجاح جاء من القاعدة، النموذج، الأداة، أو الصدفة.

23.7 تجربة الاقتصاد السياقي

يقارن استدعاء كل الوثائق بتمرير العقد المختارة عبر Vascular Context Router. النجاح: حفظ الأداء مع سياق أقل. المنهج يجب أن يتضمن baseline قويا، تثبيت النموذج قدر الإمكان، وسجل قرارات قابل للمراجعة. النتيجة التي لا تملك trace لا تصلح دليلا على النظرية؛ لأنها لا تكشف هل النجاح جاء من القاعدة، النموذج، الأداة، أو الصدفة.

23.8 تجربة التفاعل المتعدد الوكلاء

يتشارك وكلاء تخطيط وتنفيذ وتحقق HyperKB واحدة. تقاس جودة التنسيق وانخفاض التعارض بين الأدوار. المنهج يجب أن يتضمن baseline قويا، تثبيت النموذج قدر الإمكان، وسجل قرارات قابل للمراجعة. النتيجة التي لا تملك trace لا تصلح دليلا على النظرية؛ لأنها لا تكشف هل النجاح جاء من القاعدة، النموذج، الأداة، أو الصدفة.

24. مقارنة مفصلة مع الأدبيات القائمة

24.1 RAG

يعطي النموذج معرفة خارجية، لكنه غالبا لا يحول المعرفة إلى مهارة أو سياسة تشغيل. HyperKB يقبل RAG كطبقة استرجاع داخل OS أكبر. الفرق المنهجي: الورقة لا تنافس هذه الأعمال بإلغاء مكوناتها، بل تعيد ترتيبها حول مبدأ حاكم: قاعدة المعرفة كطبقة تشغيل معرفية. فإن لم يظهر هذا الترتيب في تنفيذ قابل للفحص، تسقط دعوى الخرق وتبقى الورقة مجرد تركيب اصطلاحي.

24.2 GraphRAG

يقوي العلاقات والبنى الشبكية، لكنه لا يكفي إذا لم تميز الحافة بين دليل، شرط، اعتراض، أداة، أو فشل. HyperKB تضيف وظيفة تشغيلية للحافة. الفرق المنهجي: الورقة لا تنافس هذه الأعمال بإلغاء مكوناتها، بل تعيد ترتيبها حول مبدأ حاكم: قاعدة المعرفة كطبقة تشغيل معرفية. فإن لم يظهر هذا الترتيب في تنفيذ قابل للفحص، تسقط دعوى الخرق وتبقى الورقة مجرد تركيب اصطلاحي.

24.3 ReAct

يربط التفكير بالفعل في تسلسل متداخل. HyperKB تضيف ذاكرة مهارية ومحكمة هدم وسياسة استدعاء للمعرفة التي تغذي الفعل. الفرق المنهجي: الورقة لا تنافس هذه الأعمال بإلغاء مكوناتها، بل تعيد ترتيبها حول مبدأ حاكم: قاعدة المعرفة كطبقة تشغيل معرفية. فإن لم يظهر هذا الترتيب في تنفيذ قابل للفحص، تسقط دعوى الخرق وتبقى الورقة مجرد تركيب اصطلاحي.

Toolformer 24.4

يعالج متى وكيف تستعمل النماذج الأدوات. HyperKB تضيف عقود مخاطر وسجل أثر وتقييم فشل للأداة. الفرق المنهجي: الورقة لا تنافس هذه الأعمال بإلغاء مكوناتها، بل تعيد ترتيبها حول مبدأ حاكم: قاعدة المعرفة كطبقة تشغيل معرفية. فإن لم يظهر هذا الترتيب في تنفيذ قابل للفحص، تسقط دعوى الخرق وتبقى الورقة مجرد تركيب اصطلاحي.

MemGPT 24.5

يفتح فكرة إدارة الذاكرة كأنها OS. هذه الورقة تنقل التشبيه من الذاكرة إلى قاعدة المعرفة كلها: مفاهيم، أدوات، مهارات، ومحاكم. الفرق المنهجي: الورقة لا تنافس هذه الأعمال بإلغاء مكوناتها، بل تعيد ترتيبها حول مبدأ حاكم: قاعدة المعرفة كطبقة تشغيل معرفية. فإن لم يظهر هذا الترتيب في تنفيذ قابل للفحص، تسقط دعوى الخرق وتبقى الورقة مجرد تركيب اصطلاحي.

Voyager 24.6

يظهر قيمة مكتبة مهارات قابلة للتنفيذ والتراكم. HyperKB يعمم الفكرة من بيئة Minecraft إلى أي مجال معرفي/عملي مع شروط أمان. الفرق المنهجي: الورقة لا تنافس هذه الأعمال بإلغاء مكوناتها، بل تعيد ترتيبها حول مبدأ حاكم: قاعدة المعرفة كطبقة تشغيل معرفية. فإن لم يظهر هذا الترتيب في تنفيذ قابل للفحص، تسقط دعوى الخرق وتبقى الورقة مجرد تركيب اصطلاحي.

Generative Agents 24.7

تظهر أهمية الملاحظة والذاكرة والتخطيط والتأمل. HyperKB يأخذ reflection كجزء من محكمة هدم لا كملخص ذاكرة فقط. الفرق المنهجي: الورقة لا تنافس هذه الأعمال بإلغاء مكوناتها، بل تعيد ترتيبها حول مبدأ حاكم: قاعدة المعرفة كطبقة تشغيل معرفية. فإن لم يظهر هذا الترتيب في تنفيذ قابل للفحص، تسقط دعوى الخرق وتبقى الورقة مجرد تركيب اصطلاحي.

WebArena/SWE-bench 24.8

تكشف أن المهام الواقعية الطويلة أصعب من الإجابة النصية. HyperKB يقترح مقاييس جديدة للمهام الواقعية: أثر، نقل، فشل، ضجيج، فتح سؤال. الفرق المنهجي: الورقة لا تنافس هذه الأعمال بإلغاء مكوناتها، بل تعيد ترتيبها حول مبدأ حاكم: قاعدة المعرفة كطبقة تشغيل معرفية. فإن لم يظهر هذا الترتيب في تنفيذ قابل للفحص، تسقط دعوى الخرق وتبقى الورقة مجرد تركيب اصطلاحي.

25. النسخة البرمجية الدنيا MVP

لا ينبغي أن يبدأ تنفيذ النظرية بمنصة ضخمة. النسخة الدنيا تتكون من خمسة ملفات أو جداول: جدول Claim، جدول Evidence، جدول HyperSkill، جدول Failure، وجدول Tests. فوقها يعمل Orchestrator بسيط يقرأ هدف المهمة، يستدعي البطاقات، ينفذ أداة، ثم يكتب النتيجة والفشل. معيار MVP ليس الجمال المعماري بل قدرة الباحث على تتبع لماذا استدعى النظام معلومة، لماذا استعمل أداة، ولماذا وثق أو رفض النتيجة.

مكوّن MVP 25.1

المكوّن 1 يجب أن يملك واجهة واضحة: مدخل، قرار، مخرج، وسجل. إذا كان المكوّن لا يترك أثرا قابلا للمراجعة فلا يعد جزءا من Cognitive OS. في النسخة الأولى يمكن أن تكون البطاقات YAML أو JSON، ويمكن أن تكون المحكمة قائمة أسئلة ثابتة، ويمكن أن يكون القياس يدويا. المهم أن يولد النظام بيانات تعلم حقيقية لا مجرد استعراض لغوي.

مكوّن MVP 25.2

المكوّن 2 يجب أن يملك واجهة واضحة: مدخل، قرار، مخرج، وسجل. إذا كان المكوّن لا يترك أثرا قابلا للمراجعة فلا يعد جزءا من Cognitive OS. في النسخة الأولى يمكن أن تكون البطاقات YAML أو JSON، ويمكن أن تكون المحكمة قائمة أسئلة ثابتة، ويمكن أن يكون القياس يدويا. المهم أن يولد النظام بيانات تعلم حقيقية لا مجرد استعراض لغوي.

مكوّن MVP 25.3

المكوّن 3 يجب أن يملك واجهة واضحة: مدخل، قرار، مخرج، وسجل. إذا كان المكوّن لا يترك أثرا قابلا للمراجعة فلا يعد جزءا من Cognitive OS. في النسخة الأولى يمكن أن تكون البطاقات YAML أو JSON، ويمكن أن تكون المحكمة قائمة أسئلة ثابتة، ويمكن أن يكون القياس يدويا. المهم أن يولد النظام بيانات تعلم حقيقية لا مجرد استعراض لغوي.

مكوّن MVP 25.4

المكوّن 4 يجب أن يملك واجهة واضحة: مدخل، قرار، مخرج، وسجل. إذا كان المكوّن لا يترك أثرا قابلا للمراجعة فلا يعد جزءا من Cognitive OS. في النسخة الأولى يمكن أن تكون البطاقات YAML أو JSON، ويمكن أن تكون المحكمة قائمة أسئلة ثابتة، ويمكن أن يكون القياس يدويا. المهم أن يولد النظام بيانات تعلم حقيقية لا مجرد استعراض لغوي.

مكوّن MVP 25.5

المكوّن 5 يجب أن يملك واجهة واضحة: مدخل، قرار، مخرج، وسجل. إذا كان المكوّن لا يترك أثرا قابلا للمراجعة فلا يعد جزءا من Cognitive OS. في النسخة الأولى يمكن أن تكون البطاقات YAML أو JSON، ويمكن أن تكون المحكمة قائمة أسئلة ثابتة، ويمكن أن يكون القياس يدويا. المهم أن يولد النظام بيانات تعلم حقيقية لا مجرد استعراض لغوي.

25.6 مكوّن MVP

المكوّن 6 يجب أن يملك واجهة واضحة: مدخل، قرار، مخرج، وسجل. إذا كان المكوّن لا يترك أثرا قابلا للمراجعة فلا يعد جزءا من Cognitive OS. في النسخة الأولى يمكن أن تكون البطاقات YAML أو JSON، ويمكن أن تكون المحكمة قائمة أسئلة ثابتة، ويمكن أن يكون القياس يدويا. المهم أن يولد النظام بيانات تعلم حقيقية لا مجرد استعراض لغوي.

25.7 مكوّن MVP

المكوّن 7 يجب أن يملك واجهة واضحة: مدخل، قرار، مخرج، وسجل. إذا كان المكوّن لا يترك أثرا قابلا للمراجعة فلا يعد جزءا من Cognitive OS. في النسخة الأولى يمكن أن تكون البطاقات YAML أو JSON، ويمكن أن تكون المحكمة قائمة أسئلة ثابتة، ويمكن أن يكون القياس يدويا. المهم أن يولد النظام بيانات تعلم حقيقية لا مجرد استعراض لغوي.

25.8 مكوّن MVP

المكوّن 8 يجب أن يملك واجهة واضحة: مدخل، قرار، مخرج، وسجل. إذا كان المكوّن لا يترك أثرا قابلا للمراجعة فلا يعد جزءا من Cognitive OS. في النسخة الأولى يمكن أن تكون البطاقات YAML أو JSON، ويمكن أن تكون المحكمة قائمة أسئلة ثابتة، ويمكن أن يكون القياس يدويا. المهم أن يولد النظام بيانات تعلم حقيقية لا مجرد استعراض لغوي.

25.9 مكوّن MVP

المكوّن 9 يجب أن يملك واجهة واضحة: مدخل، قرار، مخرج، وسجل. إذا كان المكوّن لا يترك أثرا قابلا للمراجعة فلا يعد جزءا من Cognitive OS. في النسخة الأولى يمكن أن تكون البطاقات YAML أو JSON، ويمكن أن تكون المحكمة قائمة أسئلة ثابتة، ويمكن أن يكون القياس يدويا. المهم أن يولد النظام بيانات تعلم حقيقية لا مجرد استعراض لغوي.

26. محكمة الهدم الرابعة: هدم فكرة الخرق نفسها

26.1 سؤال الهدم

هل يقدم هذا الإطار شيئا يتجاوز ترتيب الأدبيات؟ الجواب المشروط: نعم إذا تحقق شرط التشغيل، لا إذا بقي في مستوى التسمية. قد يقول الناقد إن كل المكونات موجودة: reflection، tool use، memory، RAG. الرد المنضبط أن الجديد لا يكون في وجود المكونات بل في جعل قاعدة المعرفة Kernel يحكم العلاقة بينها. لكن هذا الرد لا يكفي نظريا؛ يجب أن يظهر في Benchmark يفصل أثر Kernel عن أثر كل مكوّن منفرد.

26.2 سؤال الهدم

هل يقدم هذا الإطار شيئا يتجاوز ترتيب الأدبيات؟ الجواب المشروط: نعم إذا تحقق شرط التشغيل، لا إذا بقي في مستوى التسمية. قد يقول الناقد إن كل المكونات موجودة: reflection، tool use، memory، RAG. الرد المنضبط أن الجديد لا يكون في وجود المكونات بل في جعل قاعدة المعرفة Kernel يحكم العلاقة بينها. لكن هذا الرد لا يكفي نظريا؛ يجب أن يظهر في Benchmark يفصل أثر Kernel عن أثر كل مكوّن منفرد.

26.3 سؤال الهدم

هل يقدم هذا الإطار شيئا يتجاوز ترتيب الأدبيات؟ الجواب المشروط: نعم إذا تحقق شرط التشغيل، لا إذا بقي في مستوى التسمية. قد يقول الناقد إن كل المكونات موجودة: reflection، tool use، memory، RAG. الرد المنضبط أن الجديد لا يكون في وجود المكونات بل في جعل قاعدة المعرفة Kernel يحكم العلاقة بينها. لكن هذا الرد لا يكفي نظريا؛ يجب أن يظهر في Benchmark يفصل أثر Kernel عن أثر كل مكوّن منفرد.

26.4 سؤال الهدم

هل يقدم هذا الإطار شيئا يتجاوز ترتيب الأدبيات؟ الجواب المشروط: نعم إذا تحقق شرط التشغيل، لا إذا بقي في مستوى التسمية. قد يقول الناقد إن كل المكونات موجودة: reflection، tool use، memory، RAG. الرد المنضبط أن الجديد لا يكون في وجود المكونات بل في جعل قاعدة المعرفة Kernel يحكم العلاقة بينها. لكن هذا الرد لا يكفي نظريا؛ يجب أن يظهر في Benchmark يفصل أثر Kernel عن أثر كل مكوّن منفرد.

26.5 سؤال الهدم

هل يقدم هذا الإطار شيئا يتجاوز ترتيب الأدبيات؟ الجواب المشروط: نعم إذا تحقق شرط التشغيل، لا إذا بقي في مستوى التسمية. قد يقول الناقد إن كل المكونات موجودة: reflection، tool use، memory، RAG. الرد المنضبط أن الجديد لا يكون في وجود المكونات بل في جعل قاعدة المعرفة Kernel يحكم العلاقة بينها. لكن هذا الرد لا يكفي نظريا؛ يجب أن يظهر في Benchmark يفصل أثر Kernel عن أثر كل مكوّن منفرد.

26.6 سؤال الهدم

هل يقدم هذا الإطار شيئا يتجاوز ترتيب الأدبيات؟ الجواب المشروط: نعم إذا تحقق شرط التشغيل، لا إذا بقي في مستوى التسمية. قد يقول الناقد إن كل المكونات موجودة: reflection، tool use، memory، RAG. الرد المنضبط أن الجديد لا يكون في وجود المكونات بل في جعل قاعدة المعرفة Kernel يحكم العلاقة بينها. لكن هذا الرد لا يكفي نظريا؛ يجب أن يظهر في Benchmark يفصل أثر Kernel عن أثر كل مكوّن منفرد.

26.7 سؤال الهدم

هل يقدم هذا الإطار شيئا يتجاوز ترتيب الأدبيات؟ الجواب المشروط: نعم إذا تحقق شرط التشغيل، لا إذا بقي في مستوى التسمية. قد يقول الناقد إن كل المكونات موجودة: reflection، tool use، memory، RAG. الرد المنضبط أن الجديد لا يكون في وجود المكونات بل في جعل قاعدة المعرفة Kernel يحكم العلاقة بينها. لكن هذا الرد لا يكفي نظريا؛ يجب أن يظهر في Benchmark يفصل أثر Kernel عن أثر كل مكوّن منفرد.

26.8 سؤال الهدم

هل يقدم هذا الإطار شيئا يتجاوز ترتيب الأدبيات؟ الجواب المشروط: نعم إذا تحقق شرط التشغيل، لا إذا بقي في مستوى التسمية. قد يقول الناقد إن كل المكونات موجودة: reflection، tool use، memory، RAG. الرد المنضبط أن الجديد لا يكون في وجود المكونات بل في جعل قاعدة المعرفة Kernel يحكم العلاقة بينها. لكن هذا الرد لا يكفي نظريا؛ يجب أن يظهر في Benchmark يفصل أثر Kernel عن أثر كل مكوّن منفرد.

26.9 سؤال الهدم

هل يقدم هذا الإطار شيئا يتجاوز ترتيب الأدبيات؟ الجواب المشروط: نعم إذا تحقق شرط التشغيل، لا إذا بقي في مستوى التسمية. قد يقول الناقد إن كل المكونات موجودة: reflection، tool use، memory، RAG. الرد المنضبط أن الجديد لا يكون في وجود المكونات بل في جعل قاعدة المعرفة Kernel يحكم العلاقة بينها. لكن هذا الرد لا يكفي نظريا؛ يجب أن يظهر في Benchmark يفصل أثر Kernel عن أثر كل مكوّن منفرد.

26.10 سؤال الهدم

هل يقدم هذا الإطار شيئا يتجاوز ترتيب الأدبيات؟ الجواب المشروط: نعم إذا تحقق شرط التشغيل، لا إذا بقي في مستوى التسمية. قد يقول الناقد إن كل المكونات موجودة: reflection، tool use، memory، RAG. الرد المنضبط أن الجديد لا يكون في وجود المكونات بل في جعل قاعدة المعرفة Kernel يحكم العلاقة بينها. لكن هذا الرد لا يكفي نظريا؛ يجب أن يظهر في Benchmark يفصل أثر Kernel عن أثر كل مكوّن منفرد.

26.11 سؤال الهدم

هل يقدم هذا الإطار شيئا يتجاوز ترتيب الأدبيات؟ الجواب المشروط: نعم إذا تحقق شرط التشغيل، لا إذا بقي في مستوى التسمية. قد يقول الناقد إن كل المكونات موجودة: reflection، tool use، memory، RAG. الرد المنضبط أن الجديد لا يكون في وجود المكونات بل في جعل قاعدة المعرفة Kernel يحكم العلاقة بينها. لكن هذا الرد لا يكفي نظريا؛ يجب أن يظهر في Benchmark يفصل أثر Kernel عن أثر كل مكوّن منفرد.

26.12 سؤال الهدم

هل يقدم هذا الإطار شيئا يتجاوز ترتيب الأدبيات؟ الجواب المشروط: نعم إذا تحقق شرط التشغيل، لا إذا بقي في مستوى التسمية. قد يقول الناقد إن كل المكونات موجودة: reflection، tool use، memory، RAG. الرد المنضبط أن الجديد لا يكون في وجود المكونات بل في جعل قاعدة المعرفة Kernel يحكم العلاقة بينها. لكن هذا الرد لا يكفي نظريا؛ يجب أن يظهر في Benchmark يفصل أثر Kernel عن أثر كل مكوّن منفرد.

27. الصياغة الصورية الموسعة

الرمز	المعنى
K	مجموعة كتل المعرفة التشغيلية
H	مجموعة المهارات الفائقة
L	مجموعة الحلقات الوكالية
C	محاكم الهدم والتحقق
R	سجل الذاكرة وإعادة البناء
T	مجموعة الأدوات بعقود مخاطر
Q+	أسئلة الفتح الناتجة

27.1 علاقة صورية

يمكن تمثيل خطوة التشغيل بالعلاقة: $S+ \rightarrow \Delta \rightarrow K_i \rightarrow H_j \rightarrow L_t \rightarrow C_v \rightarrow R_u \rightarrow Q$. معنى هذه العلاقة أن السؤال لا ينتقل مباشرة إلى جواب، بل يمر عبر فرق فاعل يحدد ما ينبغي معرفته وما ينبغي فعله. إذا أزلنا C_v فقد نحصل على سرعة أعلى ولكن موثوقية أقل. وإذا أزلنا R_u فقد ننجح مرة ونفشل في نقل الخبرة. وإذا أزلنا $Q+$ فقد يبقى النظام ذكيا في الإجابة لكنه غير فتّاح في توسيع المجال. هذه الصياغة أولية، لكنها تمنع غموض المصطلحات لأنها تربط كل طبقة بدور تشغيلي.

27.2 علاقة صورية

يمكن تمثيل خطوة التشغيل بالعلاقة: $S+ \rightarrow \Delta \rightarrow K_i \rightarrow H_j \rightarrow L_t \rightarrow C_v \rightarrow R_u \rightarrow Q$. معنى هذه العلاقة أن السؤال لا ينتقل مباشرة إلى جواب، بل يمر عبر فرق فاعل يحدد ما ينبغي معرفته وما ينبغي فعله. إذا أزلنا C_v فقد نحصل على سرعة أعلى ولكن موثوقية أقل. وإذا أزلنا R_u فقد ننجح مرة ونفشل في نقل الخبرة. وإذا أزلنا $Q+$ فقد يبقى النظام ذكيا في الإجابة لكنه غير فتّاح في توسيع المجال. هذه الصياغة أولية، لكنها تمنع غموض المصطلحات لأنها تربط كل طبقة بدور تشغيلي.

27.3 علاقة صورية

يمكن تمثيل خطوة التشغيل بالعلاقة: $S+ \to \Delta \to K_i \to H_j \to L_t \to C_v \to R_u \to Q$. معنى هذه العلاقة أن السؤال لا ينتقل مباشرة إلى جواب، بل يمر عبر فرق فاعل يحدد ما ينبغي معرفته وما ينبغي فعله. إذا أزلنا C_v فقد نحصل على سرعة أعلى ولكن موثوقية أقل. وإذا أزلنا R_u فقد ننجح مرة ونفشل في نقل الخبرة. وإذا أزلنا Q+ فقد يبقى النظام ذكيا في الإجابة لكنه غير فتّاح في توسيع المجال. هذه الصياغة أولية، لكنها تمنع غموض المصطلحات لأنها تربط كل طبقة بدور تشغيلي.

27.4 علاقة صورية

يمكن تمثيل خطوة التشغيل بالعلاقة: $S+ \to \Delta \to K_i \to H_j \to L_t \to C_v \to R_u \to Q$. معنى هذه العلاقة أن السؤال لا ينتقل مباشرة إلى جواب، بل يمر عبر فرق فاعل يحدد ما ينبغي معرفته وما ينبغي فعله. إذا أزلنا C_v فقد نحصل على سرعة أعلى ولكن موثوقية أقل. وإذا أزلنا R_u فقد ننجح مرة ونفشل في نقل الخبرة. وإذا أزلنا Q+ فقد يبقى النظام ذكيا في الإجابة لكنه غير فتّاح في توسيع المجال. هذه الصياغة أولية، لكنها تمنع غموض المصطلحات لأنها تربط كل طبقة بدور تشغيلي.

27.5 علاقة صورية

يمكن تمثيل خطوة التشغيل بالعلاقة: $S+ \to \Delta \to K_i \to H_j \to L_t \to C_v \to R_u \to Q$. معنى هذه العلاقة أن السؤال لا ينتقل مباشرة إلى جواب، بل يمر عبر فرق فاعل يحدد ما ينبغي معرفته وما ينبغي فعله. إذا أزلنا C_v فقد نحصل على سرعة أعلى ولكن موثوقية أقل. وإذا أزلنا R_u فقد ننجح مرة ونفشل في نقل الخبرة. وإذا أزلنا Q+ فقد يبقى النظام ذكيا في الإجابة لكنه غير فتّاح في توسيع المجال. هذه الصياغة أولية، لكنها تمنع غموض المصطلحات لأنها تربط كل طبقة بدور تشغيلي.

27.6 علاقة صورية

يمكن تمثيل خطوة التشغيل بالعلاقة: $S+ \to \Delta \to K_i \to H_j \to L_t \to C_v \to R_u \to Q$. معنى هذه العلاقة أن السؤال لا ينتقل مباشرة إلى جواب، بل يمر عبر فرق فاعل يحدد ما ينبغي معرفته وما ينبغي فعله. إذا أزلنا C_v فقد نحصل على سرعة أعلى ولكن موثوقية أقل. وإذا أزلنا R_u فقد ننجح مرة ونفشل في نقل الخبرة. وإذا أزلنا Q+ فقد يبقى النظام ذكيا في الإجابة لكنه غير فتّاح في توسيع المجال. هذه الصياغة أولية، لكنها تمنع غموض المصطلحات لأنها تربط كل طبقة بدور تشغيلي.

27.7 علاقة صورية

يمكن تمثيل خطوة التشغيل بالعلاقة: $S+ \to \Delta \to K_i \to H_j \to L_t \to C_v \to R_u \to Q$. معنى هذه العلاقة أن السؤال لا ينتقل مباشرة إلى جواب، بل يمر عبر فرق فاعل يحدد ما ينبغي معرفته وما ينبغي فعله. إذا أزلنا C_v فقد نحصل على سرعة أعلى ولكن موثوقية أقل. وإذا أزلنا R_u فقد ننجح مرة ونفشل في نقل الخبرة. وإذا أزلنا Q+ فقد يبقى النظام ذكيا في الإجابة لكنه غير فتّاح في توسيع المجال. هذه الصياغة أولية، لكنها تمنع غموض المصطلحات لأنها تربط كل طبقة بدور تشغيلي.

27.8 علاقة صورية

يمكن تمثيل خطوة التشغيل بالعلاقة: $S+ \to \Delta \to K_i \to H_j \to L_t \to C_v \to R_u \to Q$. معنى هذه العلاقة أن السؤال لا ينتقل مباشرة إلى جواب، بل يمر عبر فرق فاعل يحدد ما ينبغي معرفته وما ينبغي فعله. إذا أزلنا C_v فقد نحصل على سرعة أعلى ولكن موثوقية أقل. وإذا أزلنا R_u فقد ننجح مرة ونفشل في نقل الخبرة. وإذا أزلنا Q+ فقد يبقى النظام ذكيا في الإجابة لكنه غير فتّاح في توسيع المجال. هذه الصياغة أولية، لكنها تمنع غموض المصطلحات لأنها تربط كل طبقة بدور تشغيلي.

27.9 علاقة صورية

يمكن تمثيل خطوة التشغيل بالعلاقة: $S+ \to \Delta \to K_i \to H_j \to L_t \to C_v \to R_u \to Q$. معنى هذه العلاقة أن السؤال لا ينتقل مباشرة إلى جواب، بل يمر عبر فرق فاعل يحدد ما ينبغي معرفته وما ينبغي فعله. إذا أزلنا C_v فقد نحصل على سرعة أعلى ولكن موثوقية أقل. وإذا أزلنا R_u فقد ننجح مرة ونفشل في نقل الخبرة. وإذا أزلنا Q+ فقد يبقى النظام ذكيا في الإجابة لكنه غير فتّاح في توسيع المجال. هذه الصياغة أولية، لكنها تمنع غموض المصطلحات لأنها تربط كل طبقة بدور تشغيلي.

28. مصفوفة المقارنة التفصيلية بين Agent عادي وHyperKB OS

البعد	Agent عادي	HyperKB OS	اختبار الفصل
الاسترجاع	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
الذاكرة	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
الأدوات	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
التخطيط	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
التحقق	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
التعلم من الفشل	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
نقل المجال	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
السلامة	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
فتح السؤال	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
قابلية التدقيق	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
إدارة الضجيج	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
الاقتصاد السياقي	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
تعدد النماذج	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
قابلية البحث العلمي	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن
الحدود	مكوّن منفصل أو ضمني غالبا	مكوّن محكوم بسياسة وتشغيل وسجل	اختبار ablation يفصل أثر هذا المكوّن

هذه المصفوفة تمنع خلط المقارنة. لا يجوز أن نقول إن HyperKB أفضل مطلقا، بل نسأل: في أي بعد؟ تحت أي مهمة؟ بأي تكلفة؟ وما النموذج المنافس؟ قد يفوز الوكيل العادي في السرعة والمهام القصيرة، بينما يظهر HyperKB في المهام الطويلة عالية المخاطر أو متعددة المجالات.

29. حالة تطبيقية مفصلة: تحويل كتاب أو مشروع إلى HyperKB

29.1 خطوة التحويل

تبدأ الخطوة 1 باستخراج مفهوم أو دعوى أو أداة من النص الخام. لا يُحفظ النص كما هو، بل يعاد تمثيله في بطاقة: تعريف، رتبة معرفة، آلية، مثال، اعتراض، اختبار، ومكان داخل الرسم المعرفي. بعد ذلك يربط النظام البطاقة بمهارة: تلخيص، مقارنة، قياس، تفنيد، أو توليد فرضية. النتيجة أن الكتاب لا يبقى مادة قراءة، بل يتحول إلى نظام تشغيل صغير يستطيع أن يجيب، يعترض، يقترح بحثا، ويحفظ ما تعلمه من النقد.

29.2 خطوة التحويل

تبدأ الخطوة 2 باستخراج مفهوم أو دعوى أو أداة من النص الخام. لا يُحفظ النص كما هو، بل يعاد تمثيله في بطاقة: تعريف، رتبة معرفة، آلية، مثال، اعتراض، اختبار، ومكان داخل الرسم المعرفي. بعد ذلك يربط النظام البطاقة بمهارة: تلخيص، مقارنة، قياس، تفنيد، أو توليد فرضية. النتيجة أن الكتاب لا يبقى مادة قراءة، بل يتحول إلى نظام تشغيل صغير يستطيع أن يجيب، يعترض، يقترح بحثا، ويحفظ ما تعلمه من النقد.

29.3 خطوة التحويل

تبدأ الخطوة 3 باستخراج مفهوم أو دعوى أو أداة من النص الخام. لا يُحفظ النص كما هو، بل يعاد تمثيله في بطاقة: تعريف، رتبة معرفة، آلية، مثال، اعتراض، اختبار، ومكان داخل الرسم المعرفي. بعد ذلك يربط النظام البطاقة بمهارة: تلخيص، مقارنة، قياس، تفنيد، أو توليد فرضية. النتيجة أن الكتاب لا يبقى مادة قراءة، بل يتحول إلى نظام تشغيل صغير يستطيع أن يجيب، يعترض، يقترح بحثا، ويحفظ ما تعلمه من النقد.

29.4 خطوة التحويل

تبدأ الخطوة 4 باستخراج مفهوم أو دعوى أو أداة من النص الخام. لا يُحفظ النص كما هو، بل يعاد تمثيله في بطاقة: تعريف، رتبة معرفة، آلية، مثال، اعتراض، اختبار، ومكان داخل الرسم المعرفي. بعد ذلك يربط النظام البطاقة بمهارة: تلخيص، مقارنة، قياس، تفنيد، أو توليد فرضية. النتيجة أن الكتاب لا يبقى مادة قراءة، بل يتحول إلى نظام تشغيل صغير يستطيع أن يجيب، يعترض، يقترح بحثا، ويحفظ ما تعلمه من النقد.

29.5 خطوة التحويل

تبدأ الخطوة 5 باستخراج مفهوم أو دعوى أو أداة من النص الخام. لا يُحفظ النص كما هو، بل يعاد تمثيله في بطاقة: تعريف، رتبة معرفة، آلية، مثال، اعتراض، اختبار، ومكان داخل الرسم المعرفي. بعد ذلك يربط النظام البطاقة بمهارة: تلخيص، مقارنة، قياس، تفنيد، أو توليد فرضية. النتيجة أن الكتاب لا يبقى مادة قراءة، بل يتحول إلى نظام تشغيل صغير يستطيع أن يجيب، يعترض، يقترح بحثا، ويحفظ ما تعلمه من النقد.

29.6 خطوة التحويل

تبدأ الخطوة 6 باستخراج مفهوم أو دعوى أو أداة من النص الخام. لا يُحفظ النص كما هو؛ بل يعاد تمثيله في بطاقة: تعريف، رتبة معرفة، آلية، مثال، اعتراض، اختبار، ومكان داخل الرسم المعرفي. بعد ذلك يربط النظام البطاقة بمهارة: تلخيص، مقارنة، قياس، تفنيد، أو توليد فرضية. النتيجة أن الكتاب لا يبقى مادة قراءة، بل يتحول إلى نظام تشغيل صغير يستطيع أن يجيب، يعترض، يقترح بحثا، ويحفظ ما تعلمه من النقد.

29.7 خطوة التحويل

تبدأ الخطوة 7 باستخراج مفهوم أو دعوى أو أداة من النص الخام. لا يُحفظ النص كما هو؛ بل يعاد تمثيله في بطاقة: تعريف، رتبة معرفة، آلية، مثال، اعتراض، اختبار، ومكان داخل الرسم المعرفي. بعد ذلك يربط النظام البطاقة بمهارة: تلخيص، مقارنة، قياس، تفنيد، أو توليد فرضية. النتيجة أن الكتاب لا يبقى مادة قراءة، بل يتحول إلى نظام تشغيل صغير يستطيع أن يجيب، يعترض، يقترح بحثا، ويحفظ ما تعلمه من النقد.

29.8 خطوة التحويل

تبدأ الخطوة 8 باستخراج مفهوم أو دعوى أو أداة من النص الخام. لا يُحفظ النص كما هو؛ بل يعاد تمثيله في بطاقة: تعريف، رتبة معرفة، آلية، مثال، اعتراض، اختبار، ومكان داخل الرسم المعرفي. بعد ذلك يربط النظام البطاقة بمهارة: تلخيص، مقارنة، قياس، تفنيد، أو توليد فرضية. النتيجة أن الكتاب لا يبقى مادة قراءة، بل يتحول إلى نظام تشغيل صغير يستطيع أن يجيب، يعترض، يقترح بحثا، ويحفظ ما تعلمه من النقد.

29.9 خطوة التحويل

تبدأ الخطوة 9 باستخراج مفهوم أو دعوى أو أداة من النص الخام. لا يُحفظ النص كما هو؛ بل يعاد تمثيله في بطاقة: تعريف، رتبة معرفة، آلية، مثال، اعتراض، اختبار، ومكان داخل الرسم المعرفي. بعد ذلك يربط النظام البطاقة بمهارة: تلخيص، مقارنة، قياس، تفنيد، أو توليد فرضية. النتيجة أن الكتاب لا يبقى مادة قراءة، بل يتحول إلى نظام تشغيل صغير يستطيع أن يجيب، يعترض، يقترح بحثا، ويحفظ ما تعلمه من النقد.

29.10 خطوة التحويل

تبدأ الخطوة 10 باستخراج مفهوم أو دعوى أو أداة من النص الخام. لا يُحفظ النص كما هو؛ بل يعاد تمثيله في بطاقة: تعريف، رتبة معرفة، آلية، مثال، اعتراض، اختبار، ومكان داخل الرسم المعرفي. بعد ذلك يربط النظام البطاقة بمهارة: تلخيص، مقارنة، قياس، تفنيد، أو توليد فرضية. النتيجة أن الكتاب لا يبقى مادة قراءة، بل يتحول إلى نظام تشغيل صغير يستطيع أن يجيب، يعترض، يقترح بحثا، ويحفظ ما تعلمه من النقد.

29.11 خطوة التحويل

تبدأ الخطوة 11 باستخراج مفهوم أو دعوى أو أداة من النص الخام. لا يُحفظ النص كما هو؛ بل يعاد تمثيله في بطاقة: تعريف، رتبة معرفة، آلية، مثال، اعتراض، اختبار، ومكان داخل الرسم المعرفي. بعد ذلك يربط النظام البطاقة بمهارة: تلخيص، مقارنة، قياس، تفنيد، أو توليد فرضية. النتيجة أن الكتاب لا يبقى مادة قراءة، بل يتحول إلى نظام تشغيل صغير يستطيع أن يجيب، يعترض، يقترح بحثا، ويحفظ ما تعلمه من النقد.

29.12 خطوة التحويل

تبدأ الخطوة 12 باستخراج مفهوم أو دعوى أو أداة من النص الخام. لا يُحفظ النص كما هو؛ بل يعاد تمثيله في بطاقة: تعريف، رتبة معرفة، آلية، مثال، اعتراض، اختبار، ومكان داخل الرسم المعرفي. بعد ذلك يربط النظام البطاقة بمهارة: تلخيص، مقارنة، قياس، تفنيد، أو توليد فرضية. النتيجة أن الكتاب لا يبقى مادة قراءة، بل يتحول إلى نظام تشغيل صغير يستطيع أن يجيب، يعترض، يقترح بحثا، ويحفظ ما تعلمه من النقد.

30. حالة تطبيقية مفصلة: وكيل بحث علمي فوق HyperKB

30.1 طور العمل

في الطور 1 يتعامل الوكيل مع الأدبيات لا كقائمة مراجع بل كسوق حجج. كل ورقة تدخل وفق أطروحة، دليل، حدود، وموقع في الخلاف. ثم تعمل QMCD على إبقاء النماذج المتنافسة، وتعمل محكمة الهدم على كشف المبالغة، ويعمل Fattahic Question Engine على إنتاج سؤال بحثي جديد. معيار النجاح ليس عدد المراجع، بل قدرة النظام على توليد فرضية محددة يمكن اختبارها وتحديد ما الذي سيكسرها.

30.2 طور العمل

في الطور 2 يتعامل الوكيل مع الأدبيات لا كقائمة مراجع بل كسوق حجج. كل ورقة تدخل وفق أطروحة، دليل، حدود، وموقع في الخلاف. ثم تعمل QMCD على إبقاء النماذج المتنافسة، وتعمل محكمة الهدم على كشف المبالغة، ويعمل Fattahic Question Engine على إنتاج سؤال بحثي جديد. معيار النجاح ليس عدد المراجع، بل قدرة النظام على توليد فرضية محددة يمكن اختبارها وتحديد ما الذي سيكسرها.

30.3 طور العمل

في الطور 3 يتعامل الوكيل مع الأدبيات لا كقائمة مراجع بل كسوق حجج. كل ورقة تدخل وفق أطروحة، دليل، حدود، وموقع في الخلاف. ثم تعمل QMCD على إبقاء النماذج المتنافسة، وتعمل محكمة الهدم على كشف المبالغة، ويعمل Fattahic Question Engine على إنتاج سؤال بحثي جديد. معيار النجاح ليس عدد المراجع، بل قدرة النظام على توليد فرضية محددة يمكن اختبارها وتحديد ما الذي سيكسرها.

30.4 طور العمل

في الطور 4 يتعامل الوكيل مع الأدبيات لا كقائمة مراجع بل كسوق حجج. كل ورقة تدخل وفق أطروحة، دليل، حدود، وموقع في الخلاف. ثم تعمل QMCD على إبقاء النماذج المتنافسة، وتعمل محكمة الهدم على كشف المبالغة، ويعمل Fattahic Question Engine على إنتاج سؤال بحثي جديد. معيار النجاح ليس عدد المراجع، بل قدرة النظام على توليد فرضية محددة يمكن اختبارها وتحديد ما الذي سيكسرها.

30.5 طور العمل

في الطور 5 يتعامل الوكيل مع الأدبيات لا كقائمة مراجع بل كسوق حجج. كل ورقة تدخل وفق أطروحة، دليل، حدود، وموقع في الخلاف. ثم تعمل QMCD على إبقاء النماذج المتنافسة، وتعمل محكمة الهدم على كشف المبالغة، ويعمل Fattahic Question Engine على إنتاج سؤال بحثي جديد. معيار النجاح ليس عدد المراجع، بل قدرة النظام على توليد فرضية محددة يمكن اختبارها وتحديد ما الذي سيكسرها.

30.6 طور العمل

في الطور 6 يتعامل الوكيل مع الأدبيات لا كقائمة مراجع بل كسوق حجج. كل ورقة تدخل وفق أطروحة، دليل، حدود، وموقع في الخلاف. ثم تعمل QMCD على إبقاء النماذج المتنافسة، وتعمل محكمة الهدم على كشف المبالغة، ويعمل Fattahic Question Engine على إنتاج سؤال بحثي جديد. معيار النجاح ليس عدد المراجع، بل قدرة النظام على توليد فرضية محددة يمكن اختبارها وتحديد ما الذي سيكسرها.

30.7 طور العمل

في الطور 7 يتعامل الوكيل مع الأدبيات لا كقائمة مراجع بل كسوق حجج. كل ورقة تدخل وفق أطروحة، دليل، حدود، وموقع في الخلاف. ثم تعمل QMCD على إبقاء النماذج المتنافسة، وتعمل محكمة الهدم على كشف المبالغة، ويعمل Fattahic Question Engine على إنتاج سؤال بحثي جديد. معيار النجاح ليس عدد المراجع، بل قدرة النظام على توليد فرضية محددة يمكن اختبارها وتحديد ما الذي سيكسرها.

30.8 طور العمل

في الطور 8 يتعامل الوكيل مع الأدبيات لا كقائمة مراجع بل كسوق حجج. كل ورقة تدخل وفق أطروحة، دليل، حدود، وموقع في الخلاف. ثم تعمل QMCD على إبقاء النماذج المتنافسة، وتعمل محكمة الهدم على كشف المبالغة، ويعمل Fattahic Question Engine على إنتاج سؤال بحثي جديد. معيار النجاح ليس عدد المراجع، بل قدرة النظام على توليد فرضية محددة يمكن اختبارها وتحديد ما الذي سيكسرها.

30.9 طور العمل

في الطور 9 يتعامل الوكيل مع الأدبيات لا كقائمة مراجع بل كسوق حجج. كل ورقة تدخل وفق أطروحة، دليل، حدود، وموقع في الخلاف. ثم تعمل QMCD على إبقاء النماذج المتنافسة، وتعمل محكمة الهدم على كشف المبالغة، ويعمل Fattahic Question Engine على إنتاج سؤال بحثي جديد. معيار النجاح ليس عدد المراجع، بل قدرة النظام على توليد فرضية محددة يمكن اختبارها وتحديد ما الذي سيكسرها.

30.10 طور العمل

في الطور 10 يتعامل الوكيل مع الأدبيات لا كقائمة مراجع بل كسوق حجج. كل ورقة تدخل وفق أطروحة، دليل، حدود، وموقع في الخلاف. ثم تعمل QMCD على إبقاء النماذج المتنافسة، وتعمل محكمة الهدم على كشف المبالغة، ويعمل Fattahic Question Engine على إنتاج سؤال بحثي جديد. معيار النجاح ليس عدد المراجع، بل قدرة النظام على توليد فرضية محددة يمكن اختبارها وتحديد ما الذي سيكسرها.

31. حالة تطبيقية مفصلة: وكيل برمجي وإصلاح أخطاء

31.1 دورة إصلاح

تبدأ دورة الإصلاح بقراءة issue واستخراج Δ: هل الخلل في منطق، واجهة، اختبار، تبعية، أو فهم مطلب؟ ثم يستدعي HyperKB خبرات إصلاح سابقة لا كأمثلة نصية فقط بل كمهارات: تحديد ملف، بناء اختبار، تعديل، تشغيل، ومراجعة. إذا فشل الاختبار، لا تضيع التجربة؛ تُكتب في Failure Ledger وتتحول إلى قيد في الدورة التالية. هذا هو الفرق بين وكيل يكرر patch عشوائيا ووكيل يطور سياسة إصلاح.

31.2 دورة إصلاح

تبدأ دورة الإصلاح بقراءة issue واستخراج Δ: هل الخلل في منطق، واجهة، اختبار، تبعية، أو فهم مطلب؟ ثم يستدعي HyperKB خبرات إصلاح سابقة لا كأمثلة نصية فقط بل كمهارات: تحديد ملف، بناء اختبار، تعديل، تشغيل، ومراجعة. إذا فشل الاختبار، لا تضيع التجربة؛ تُكتب في Failure Ledger وتتحول إلى قيد في الدورة التالية. هذا هو الفرق بين وكيل يكرر patch عشوائيا ووكيل يطور سياسة إصلاح.

31.3 دورة إصلاح

تبدأ دورة الإصلاح بقراءة issue واستخراج Δ: هل الخلل في منطق، واجهة، اختبار، تبعية، أو فهم مطلب؟ ثم يستدعي HyperKB خبرات إصلاح سابقة لا كأمثلة نصية فقط بل كمهارات: تحديد ملف، بناء اختبار، تعديل، تشغيل، ومراجعة. إذا فشل الاختبار، لا تضيع التجربة؛ تُكتب في Failure Ledger وتتحول إلى قيد في الدورة التالية. هذا هو الفرق بين وكيل يكرر patch عشوائيا ووكيل يطور سياسة إصلاح.

31.4 دورة إصلاح

تبدأ دورة الإصلاح بقراءة issue واستخراج Δ: هل الخلل في منطق، واجهة، اختبار، تبعية، أو فهم مطلب؟ ثم يستدعي HyperKB خبرات إصلاح سابقة لا كأمثلة نصية فقط بل كمهارات: تحديد ملف، بناء اختبار، تعديل، تشغيل، ومراجعة. إذا فشل الاختبار، لا تضيع التجربة؛ تُكتب في Failure Ledger وتتحول إلى قيد في الدورة التالية. هذا هو الفرق بين وكيل يكرر patch عشوائيا ووكيل يطور سياسة إصلاح.

31.5 دورة إصلاح

تبدأ دورة الإصلاح بقراءة issue واستخراج Δ: هل الخلل في منطق، واجهة، اختبار، تبعية، أو فهم مطلب؟ ثم يستدعي HyperKB خبرات إصلاح سابقة لا كأمثلة نصية فقط بل كمهارات: تحديد ملف، بناء اختبار، تعديل، تشغيل، ومراجعة. إذا فشل الاختبار، لا تضيع التجربة؛ تُكتب في Failure Ledger وتتحول إلى قيد في الدورة التالية. هذا هو الفرق بين وكيل يكرر patch عشوائيا ووكيل يطور سياسة إصلاح.

31.6 دورة إصلاح

تبدأ دورة الإصلاح بقراءة issue واستخراج Δ: هل الخلل في منطق، واجهة، اختبار، تبعية، أو فهم مطلب؟ ثم يستدعي HyperKB خبرات إصلاح سابقة لا كأمثلة نصية فقط بل كمهارات: تحديد ملف، بناء اختبار، تعديل، تشغيل، ومراجعة. إذا فشل الاختبار، لا تضيع التجربة؛ تُكتب في Failure Ledger وتتحول إلى قيد في الدورة التالية. هذا هو الفرق بين وكيل يكرر patch عشوائيا ووكيل يطور سياسة إصلاح.

31.7 دورة إصلاح

تبدأ دورة الإصلاح بقراءة issue واستخراج Δ: هل الخلل في منطق، واجهة، اختبار، تبعية، أو فهم مطلب؟ ثم يستدعي HyperKB خبرات إصلاح سابقة لا كأمثلة نصية فقط بل كمهارات: تحديد ملف، بناء اختبار، تعديل، تشغيل، ومراجعة. إذا فشل الاختبار، لا تضيع التجربة؛ تُكتب في Failure Ledger وتتحول إلى قيد في الدورة التالية. هذا هو الفرق بين وكيل يكرر patch عشوائيا ووكيل يطور سياسة إصلاح.

31.8 دورة إصلاح

تبدأ دورة الإصلاح بقراءة issue واستخراج Δ: هل الخلل في منطق، واجهة، اختبار، تبعية، أو فهم مطلب؟ ثم يستدعي HyperKB خبرات إصلاح سابقة لا كأمثلة نصية فقط بل كمهارات: تحديد ملف، بناء اختبار، تعديل، تشغيل، ومراجعة. إذا فشل الاختبار، لا تضيع التجربة؛ تُكتب في Failure Ledger وتتحول إلى قيد في الدورة التالية. هذا هو الفرق بين وكيل يكرر patch عشوائيا ووكيل يطور سياسة إصلاح.

31.9 دورة إصلاح

تبدأ دورة الإصلاح بقراءة issue واستخراج Δ: هل الخلل في منطق، واجهة، اختبار، تبعية، أو فهم مطلب؟ ثم يستدعي HyperKB خبرات إصلاح سابقة لا كأمثلة نصية فقط بل كمهارات: تحديد ملف، بناء اختبار، تعديل، تشغيل، ومراجعة. إذا فشل الاختبار، لا تضيع التجربة؛ تُكتب في Failure Ledger وتتحول إلى قيد في الدورة التالية. هذا هو الفرق بين وكيل يكرر patch عشوائيا ووكيل يطور سياسة إصلاح.

31.10 دورة إصلاح

تبدأ دورة الإصلاح بقراءة issue واستخراج Δ: هل الخلل في منطق، واجهة، اختبار، تبعية، أو فهم مطلب؟ ثم يستدعي HyperKB خبرات إصلاح سابقة لا كأمثلة نصية فقط بل كمهارات: تحديد ملف، بناء اختبار، تعديل، تشغيل، ومراجعة. إذا فشل الاختبار، لا تضيع التجربة؛ تُكتب في Failure Ledger وتتحول إلى قيد في الدورة التالية. هذا هو الفرق بين وكيل يكرر patch عشوائيا ووكيل يطور سياسة إصلاح.

32. مخاطر السلامة والأمن المعرفي

32.1 حقن التعليمات داخل الوثائق

الخطر هنا لا يعالج بنصيحة عامة، بل بآلية داخل OS: تصنيف المصدر، عقد الأداة، حدود الصلاحية، سجل القرار، ومراجعة دورية للذاكرة. إذا لم تُبن هذه الآليات، يمكن أن تصبح HyperKB أخطر من RAG عادي لأنها تمنح النظام ثقة تنظيمية زائفة. لذلك السلامة ليست ملحقا؛ إنها جزء من الحاكمية المعرفية.

32.2 تضارب صلاحيات الأدوات

الخطر هنا لا يعالج بنصيحة عامة، بل بآلية داخل OS: تصنيف المصدر، عقد الأداة، حدود الصلاحية، سجل القرار، ومراجعة دورية للذاكرة. إذا لم تُبن هذه الآليات، يمكن أن تصبح HyperKB أخطر من RAG عادي لأنها تمنح النظام ثقة تنظيمية زائفة. لذلك السلامة ليست ملحقا؛ إنها جزء من الحاكمية المعرفية.

32.3 تسريب بيانات من ذاكرة طويلة

الخطر هنا لا يعالج بنصيحة عامة، بل بآلية داخل OS: تصنيف المصدر، عقد الأداة، حدود الصلاحية، سجل القرار، ومراجعة دورية للذاكرة. إذا لم تُبن هذه الآليات، يمكن أن تصبح HyperKB أخطر من RAG عادي لأنها تمنح النظام ثقة تنظيمية زائفة. لذلك السلامة ليست ملحقا؛ إنها جزء من الحاكمية المعرفية.

32.4 الحلقات اللانهائية

الخطر هنا لا يعالج بنصيحة عامة، بل بآلية داخل OS: تصنيف المصدر، عقد الأداة، حدود الصلاحية، سجل القرار، ومراجعة دورية للذاكرة. إذا لم تُبن هذه الآليات، يمكن أن تصبح HyperKB أخطر من RAG عادي لأنها تمنح النظام ثقة تنظيمية زائفة. لذلك السلامة ليست ملحقا؛ إنها جزء من الحاكمية المعرفية.

32.5 الثقة الزائدة بعد نجاح سابق

الخطر هنا لا يعالج بنصيحة عامة، بل بآلية داخل OS: تصنيف المصدر، عقد الأداة، حدود الصلاحية، سجل القرار، ومراجعة دورية للذاكرة. إذا لم تُبن هذه الآليات، يمكن أن تصبح HyperKB أخطر من RAG عادي لأنها تمنح النظام ثقة تنظيمية زائفة. لذلك السلامة ليست ملحقا؛ إنها جزء من الحاكمية المعرفية.

32.6 تسميم قاعدة المعرفة

الخطر هنا لا يعالج بنصيحة عامة، بل بآلية داخل OS: تصنيف المصدر، عقد الأداة، حدود الصلاحية، سجل القرار، ومراجعة دورية للذاكرة. إذا لم تُبن هذه الآليات، يمكن أن تصبح HyperKB أخطر من RAG عادي لأنها تمنح النظام ثقة تنظيمية زائفة. لذلك السلامة ليست ملحقا؛ إنها جزء من الحاكمية المعرفية.

32.7 خلط المصدر الضعيف بالقوي

الخطر هنا لا يعالج بنصيحة عامة، بل بآلية داخل OS: تصنيف المصدر، عقد الأداة، حدود الصلاحية، سجل القرار، ومراجعة دورية للذاكرة. إذا لم تُبن هذه الآليات، يمكن أن تصبح HyperKB أخطر من RAG عادي لأنها تمنح النظام ثقة تنظيمية زائفة. لذلك السلامة ليست ملحقا؛ إنها جزء من الحاكمية المعرفية.

32.8 تحويل الاستعارة إلى آلية مزيفة

الخطر هنا لا يعالج بنصيحة عامة، بل بآلية داخل OS: تصنيف المصدر، عقد الأداة، حدود الصلاحية، سجل القرار، ومراجعة دورية للذاكرة. إذا لم تُبن هذه الآليات، يمكن أن تصبح HyperKB أخطر من RAG عادي لأنها تمنح النظام ثقة تنظيمية زائفة. لذلك السلامة ليست ملحقا؛ إنها جزء من الحاكمية المعرفية.

32.9 تجاوز المستخدم أو المجال

الخطر هنا لا يعالج بنصيحة عامة، بل بآلية داخل OS: تصنيف المصدر، عقد الأداة، حدود الصلاحية، سجل القرار، ومراجعة دورية للذاكرة. إذا لم تُبن هذه الآليات، يمكن أن تصبح HyperKB أخطر من RAG عادي لأنها تمنح النظام ثقة تنظيمية زائفة. لذلك السلامة ليست ملحقا؛ إنها جزء من الحاكمية المعرفية.

32.10 الاعتماد على Benchmark ملوث

الخطر هنا لا يعالج بنصيحة عامة، بل بآلية داخل OS: تصنيف المصدر، عقد الأداة، حدود الصلاحية، سجل القرار، ومراجعة دورية للذاكرة. إذا لم تُبن هذه الآليات، يمكن أن تصبح HyperKB أخطر من RAG عادي لأنها تمنح النظام ثقة تنظيمية زائفة. لذلك السلامة ليست ملحقا؛ إنها جزء من الحاكمية المعرفية.

33. محاور بحثية إضافية بعيدة المدى

33.1 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط؛ لذلك يجب تقييم الفرضيات خارجيا.

33.2 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط؛ لذلك يجب تقييم الفرضيات خارجيا.

33.3 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط؛ لذلك يجب تقييم الفرضيات خارجيا.

33.4 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط؛ لذلك يجب تقييم الفرضيات خارجيا.

33.5 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط؛ لذلك يجب تقييم الفرضيات خارجيا.

33.6 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط؛ لذلك يجب تقييم الفرضيات خارجيا.

33.7 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط؛ لذلك يجب تقييم الفرضيات خارجيا.

33.8 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط؛ لذلك يجب تقييم الفرضيات خارجيا.

33.9 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط؛ لذلك يجب تقييم الفرضيات خارجيا.

33.10 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط؛ لذلك يجب تقييم الفرضيات خارجيا.

33.11 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط؛ لذلك يجب تقييم الفرضيات خارجيا.

33.12 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط؛ لذلك يجب تقييم الفرضيات خارجيا.

33.13 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط، لذلك يجب تقييم الفرضيات خارجيا.

33.14 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط، لذلك يجب تقييم الفرضيات خارجيا.

33.15 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط، لذلك يجب تقييم الفرضيات خارجيا.

33.16 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط، لذلك يجب تقييم الفرضيات خارجيا.

33.17 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط، لذلك يجب تقييم الفرضيات خارجيا.

33.18 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط، لذلك يجب تقييم الفرضيات خارجيا.

33.19 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط، لذلك يجب تقييم الفرضيات خارجيا.

33.20 محور بعيد

العنوان: اختبار قدرة Cognitive OS على فتح مجال معرفي جديد في حالة نقص البيانات. السؤال: هل يستطيع النظام أن يميز بين ما يعرفه وما يجهله، ثم يحول الجهل إلى برنامج جمع بيانات؟ الفرضية: وجود سلم مراتب معرفة ومحكمة هدم يرفع جودة الاعتراف بالنقص. المنهج: مقارنة إجابات في مجال جديد مع خبراء، ثم قياس عدد الادعاءات الزائدة وعدد الفرضيات القابلة للاختبار. الاعتراض: ربما ينتج النظام لغة بحثية فقط، لذلك يجب تقييم الفرضيات خارجيا.

34. نسخة تنفيذية مختصرة للباحثين والمهندسين

لتحويل هذه الورقة إلى مشروع، ابدأ بخمسين بطاقة معرفة، عشر مهارات، ثلاث أدوات، محكمة هدم من عشرين سؤالا، ومؤشرين: KOS-I وDRI. لا تبدأ بمنصة ضخمة، ابدأ بمهمة محددة يمكن قياسها. اجعل كل قرار قابلا للتدقيق: لماذا اخترت هذه المعرفة؟ لماذا استعملت هذه الأداة؟ ما الاعتراض؟ ماذا تغير بعد الفشل؟ ثم انشر النتائج حتى لو كانت سلبية، لأن النظرية لا تنضج بالنجاح فقط بل بسجل الهدم.

34.1 قاعدة تنفيذية

القاعدة 1: لا تضف طبقة إلا إذا كانت تخفض خطرا أو ترفع نقلا أو تزيد قابلية التحقق. كل طبقة تزيد التعقيد يجب أن تدفع ثمنها بأثر قابل للقياس. وإذا لم تستطع قياس الأثر، فصنف الطبقة كفرضية هندسية لا كمبدأ مثبت. بهذا الانضباط يبقى الإطار خرقيا دون أن يسقط في التضخم.

34.2 قاعدة تنفيذية

القاعدة 2: لا تضف طبقة إلا إذا كانت تخفض خطرا أو ترفع نقلا أو تزيد قابلية التحقق. كل طبقة تزيد التعقيد يجب أن تدفع ثمنها بأثر قابل للقياس. وإذا لم تستطع قياس الأثر، فصنف الطبقة كفرضية هندسية لا كمبدأ مثبت. بهذا الانضباط يبقى الإطار خرقيا دون أن يسقط في التضخم.

34.3 قاعدة تنفيذية

القاعدة 3: لا تضف طبقة إلا إذا كانت تخفض خطرا أو ترفع نقلا أو تزيد قابلية التحقق. كل طبقة تزيد التعقيد يجب أن تدفع ثمنها بأثر قابل للقياس. وإذا لم تستطع قياس الأثر، فصنف الطبقة كفرضية هندسية لا كمبدأ مثبت. بهذا الانضباط يبقى الإطار خرقيا دون أن يسقط في التضخم.

34.4 قاعدة تنفيذية

القاعدة 4: لا تضف طبقة إلا إذا كانت تخفض خطرا أو ترفع نقلا أو تزيد قابلية التحقق. كل طبقة تزيد التعقيد يجب أن تدفع ثمنها بأثر قابل للقياس. وإذا لم تستطع قياس الأثر، فصنف الطبقة كفرضية هندسية لا كمبدأ مثبت. بهذا الانضباط يبقى الإطار خرقيا دون أن يسقط في التضخم.

34.5 قاعدة تنفيذية

القاعدة 5: لا تضف طبقة إلا إذا كانت تخفض خطرا أو ترفع نقلا أو تزيد قابلية التحقق. كل طبقة تزيد التعقيد يجب أن تدفع ثمنها بأثر قابل للقياس. وإذا لم تستطع قياس الأثر، فصنف الطبقة كفرضية هندسية لا كمبدأ مثبت. بهذا الانضباط يبقى الإطار خرقيا دون أن يسقط في التضخم.

34.6 قاعدة تنفيذية

القاعدة 6: لا تضف طبقة إلا إذا كانت تخفض خطرا أو ترفع نقلا أو تزيد قابلية التحقق. كل طبقة تزيد التعقيد يجب أن تدفع ثمنها بأثر قابل للقياس. وإذا لم تستطع قياس الأثر، فصنف الطبقة كفرضية هندسية لا كمبدأ مثبت. بهذا الانضباط يبقى الإطار خرقيا دون أن يسقط في التضخم.

34.7 قاعدة تنفيذية

القاعدة 7: لا تضف طبقة إلا إذا كانت تخفض خطرا أو ترفع نقلا أو تزيد قابلية التحقق. كل طبقة تزيد التعقيد يجب أن تدفع ثمنها بأثر قابل للقياس. وإذا لم تستطع قياس الأثر، فصنف الطبقة كفرضية هندسية لا كمبدأ مثبت. بهذا الانضباط يبقى الإطار خرقيا دون أن يسقط في التضخم.

34.8 قاعدة تنفيذية

القاعدة 8: لا تضف طبقة إلا إذا كانت تخفض خطرا أو ترفع نقلا أو تزيد قابلية التحقق. كل طبقة تزيد التعقيد يجب أن تدفع ثمنها بأثر قابل للقياس. وإذا لم تستطع قياس الأثر، فصنف الطبقة كفرضية هندسية لا كمبدأ مثبت. بهذا الانضباط يبقى الإطار خرقيا دون أن يسقط في التضخم.

34.9 قاعدة تنفيذية

القاعدة 9: لا تضف طبقة إلا إذا كانت تخفض خطرا أو ترفع نقلا أو تزيد قابلية التحقق. كل طبقة تزيد التعقيد يجب أن تدفع ثمنها بأثر قابل للقياس. وإذا لم تستطع قياس الأثر، فصنف الطبقة كفرضية هندسية لا كمبدأ مثبت. بهذا الانضباط يبقى الإطار خرقيا دون أن يسقط في التضخم.

ملحق أ: مصفوفة تحويل الفكرة إلى كتلة معرفة تشغيلية

أ.1 بطاقة تشغيل نموذجية

تبدأ البطاقة باسم المفهوم، ثم تعريفه، ثم سبب الحاجة إليه، ثم علاقة المفهوم بالمهمة، ثم الأدوات التي تستعمله، ثم شروط الحدود، ثم اعتراضه الأقوى، ثم اختبار قبول. الهدف من هذه البطاقة منع تحول القاعدة المعرفية إلى مخزن نصوص. في إطار HyperKB، كل بطاقة يجب أن تجيب: متى تُستدعى؟ ماذا تفعل؟ ماذا تمنع؟ كيف تفشل؟ وكيف تنقل إلى مجال آخر؟

أ.2 بطاقة تشغيل نموذجية

تبدأ البطاقة باسم المفهوم، ثم تعريفه، ثم سبب الحاجة إليه، ثم علاقة المفهوم بالمهمة، ثم الأدوات التي تستعمله، ثم شروط الحدود، ثم اعتراضه الأقوى، ثم اختبار قبول. الهدف من هذه البطاقة منع تحول القاعدة المعرفية إلى مخزن نصوص. في إطار HyperKB، كل بطاقة يجب أن تجيب: متى تُستدعى؟ ماذا تفعل؟ ماذا تمنع؟ كيف تفشل؟ وكيف تنقل إلى مجال آخر؟

أ.3 بطاقة تشغيل نموذجية

تبدأ البطاقة باسم المفهوم، ثم تعريفه، ثم سبب الحاجة إليه، ثم علاقة المفهوم بالمهمة، ثم الأدوات التي تستعمله، ثم شروط الحدود، ثم اعتراضه الأقوى، ثم اختبار قبول. الهدف من هذه البطاقة منع تحول القاعدة المعرفية إلى مخزن نصوص. في إطار HyperKB، كل بطاقة يجب أن تجيب: متى تُستدعى؟ ماذا تفعل؟ ماذا تمنع؟ كيف تفشل؟ وكيف تنقل إلى مجال آخر؟

أ.4 بطاقة تشغيل نموذجية

تبدأ البطاقة باسم المفهوم، ثم تعريفه، ثم سبب الحاجة إليه، ثم علاقة المفهوم بالمهمة، ثم الأدوات التي تستعمله، ثم شروط الحدود، ثم اعتراضه الأقوى، ثم اختبار قبول. الهدف من هذه البطاقة منع تحول القاعدة المعرفية إلى مخزن نصوص. في إطار HyperKB، كل بطاقة يجب أن تجيب: متى تُستدعى؟ ماذا تفعل؟ ماذا تمنع؟ كيف تفشل؟ وكيف تنقل إلى مجال آخر؟

أ.5 بطاقة تشغيل نموذجية

تبدأ البطاقة باسم المفهوم، ثم تعريفه، ثم سبب الحاجة إليه، ثم علاقة المفهوم بالمهمة، ثم الأدوات التي تستعمله، ثم شروط الحدود، ثم اعتراضه الأقوى، ثم اختبار قبول. الهدف من هذه البطاقة منع تحول القاعدة المعرفية إلى مخزن نصوص. في إطار HyperKB، كل بطاقة يجب أن تجيب: متى تُستدعى؟ ماذا تفعل؟ ماذا تمنع؟ كيف تفشل؟ وكيف تنقل إلى مجال آخر؟

أ.6 بطاقة تشغيل نموذجية

تبدأ البطاقة باسم المفهوم، ثم تعريفه، ثم سبب الحاجة إليه، ثم علاقة المفهوم بالمهمة، ثم الأدوات التي تستعمله، ثم شروط الحدود، ثم اعتراضه الأقوى، ثم اختبار قبول. الهدف من هذه البطاقة منع تحول القاعدة المعرفية إلى مخزن نصوص. في إطار HyperKB، كل بطاقة يجب أن تجيب: متى تُستدعى؟ ماذا تفعل؟ ماذا تمنع؟ كيف تفشل؟ وكيف تنقل إلى مجال آخر؟

أ.7 بطاقة تشغيل نموذجية

تبدأ البطاقة باسم المفهوم، ثم تعريفه، ثم سبب الحاجة إليه، ثم علاقة المفهوم بالمهمة، ثم الأدوات التي تستعمله، ثم شروط الحدود، ثم اعتراضه الأقوى، ثم اختبار قبول. الهدف من هذه البطاقة منع تحول القاعدة المعرفية إلى مخزن نصوص. في إطار HyperKB، كل بطاقة يجب أن تجيب: متى تُستدعى؟ ماذا تفعل؟ ماذا تمنع؟ كيف تفشل؟ وكيف تنقل إلى مجال آخر؟

أ.8 بطاقة تشغيل نموذجية

تبدأ البطاقة باسم المفهوم، ثم تعريفه، ثم سبب الحاجة إليه، ثم علاقة المفهوم بالمهمة، ثم الأدوات التي تستعمله، ثم شروط الحدود، ثم اعتراضه الأقوى، ثم اختبار قبول. الهدف من هذه البطاقة منع تحول القاعدة المعرفية إلى مخزن نصوص. في إطار HyperKB، كل بطاقة يجب أن تجيب: متى تُستدعى؟ ماذا تفعل؟ ماذا تمنع؟ كيف تفشل؟ وكيف تنقل إلى مجال آخر؟

أ.9 بطاقة تشغيل نموذجية

تبدأ البطاقة باسم المفهوم، ثم تعريفه، ثم سبب الحاجة إليه، ثم علاقة المفهوم بالمهمة، ثم الأدوات التي تستعمله، ثم شروط الحدود، ثم اعتراضه الأقوى، ثم اختبار قبول. الهدف من هذه البطاقة منع تحول القاعدة المعرفية إلى مخزن نصوص. في إطار HyperKB، كل بطاقة يجب أن تجيب: متى تُستدعى؟ ماذا تفعل؟ ماذا تمنع؟ كيف تفشل؟ وكيف تنقل إلى مجال آخر؟

أ.10 بطاقة تشغيل نموذجية

تبدأ البطاقة باسم المفهوم، ثم تعريفه، ثم سبب الحاجة إليه، ثم علاقة المفهوم بالمهمة، ثم الأدوات التي تستعمله، ثم شروط الحدود، ثم اعتراضه الأقوى، ثم اختبار قبول. الهدف من هذه البطاقة منع تحول القاعدة المعرفية إلى مخزن نصوص. في إطار HyperKB، كل بطاقة يجب أن تجيب: متى تُستدعى؟ ماذا تفعل؟ ماذا تمنع؟ كيف تفشل؟ وكيف تنقل إلى مجال آخر؟

أ.11 بطاقة تشغيل نموذجية

تبدأ البطاقة باسم المفهوم، ثم تعريفه، ثم سبب الحاجة إليه، ثم علاقة المفهوم بالمهمة، ثم الأدوات التي تستعمله، ثم شروط الحدود، ثم اعتراضه الأقوى، ثم اختبار قبول. الهدف من هذه البطاقة منع تحول القاعدة المعرفية إلى مخزن نصوص. في إطار HyperKB، كل بطاقة يجب أن تجيب: متى تُستدعى؟ ماذا تفعل؟ ماذا تمنع؟ كيف تفشل؟ وكيف تنقل إلى مجال آخر؟

أ.12 بطاقة تشغيل نموذجية

تبدأ البطاقة باسم المفهوم، ثم تعريفه، ثم سبب الحاجة إليه، ثم علاقة المفهوم بالمهمة، ثم الأدوات التي تستعمله، ثم شروط الحدود، ثم اعتراضه الأقوى، ثم اختبار قبول. الهدف من هذه البطاقة منع تحول القاعدة المعرفية إلى مخزن نصوص. في إطار HyperKB، كل بطاقة يجب أن تجيب: متى تُستدعى؟ ماذا تفعل؟ ماذا تمنع؟ كيف تفشل؟ وكيف تنقل إلى مجال آخر؟

ملحق ب: محاور بحثية تفصيلية

ب.1 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.2 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.3 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.4 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.5 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.6 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.7 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.8 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.9 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.10 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.11 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.12 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.13 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.14 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.15 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.16 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.17 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.18 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.19 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ب.20 محور بحثي

السؤال: كيف يؤثر مكوّن معين من Cognitive OS في نجاح مهمة طويلة؟ الفرضية: كلما زادت صراحة سياسة الاستدعاء والتحقق، انخفض الفشل المتكرر. المنهج: مقارنة تجريبية بين baseline ونسخة HyperKB مع تثبيت النموذج والأدوات. البيانات: سجلات تنفيذ، درجات نجاح، أخطاء متكررة، وأسئلة لاحقة. الاعتراض المتوقع: قد يكون التحسن بسبب prompt أطول لا بسبب OS؛ ويجاب عنه بتثبيت طول السياق واختبار ablation.

ملحق ج: قاموس اصطلاحي مركز

المصطلح	الوظيفة
نواة مساعف للتشغيل المعرفي	Kernel يحدد سياسات استدعاء المعرفة والأدوات والاختبارات.
خلية المهارة الفائقة	وحدة معرفة-أداة-اختبار-ذاكرة قابلة لإعادة الاستعمال.
الحلقة الوكالية الفائقة	حلقة تخطيط وتنفيذ وهدم وتحديث لا تكرر فقط بل تتعلم.
كُلّة معرفة بمستوى AGI	معرفة لها تعريف، شروط، نقل، اختبار، وحدود.
فتحة السؤال الفتّاح	آلية توليد سؤال أعلى بعد حل أو فشل.
بوابة الهدم الميتا-معرفي	شرط نقدي قبل اعتماد نتيجة أو فعل.
الموجّه الوعائي للسياق	توجيه موارد السياق إلى العقد الأكثر أثرا.
مخطط الوكالة القابل للنقل	قالب يحفظ بنية الفعل لا نص المجال.
نظرية بطاقة المعرفة التشغيلية	كل وثيقة تتحول إلى بطاقة تحتوي وظيفة استعمال وحدودا واختبارا.
نظرية سجل الفشل الوكالي	الفشل مورد معرفي يحدّث السياسة المستقبلية.
نظرية عتبة التحول من KB إلى OS	قاعدة المعرفة لا تصير OS إلا حين تملك Kernel وScheduler وVerifier.
نظرية النقل البنيوي	النقل بين المجالات يتم بنقل علاقات لا نقل جمل.
نظرية الضجيج المعرفي السلبي	ازدياد المعرفة غير المحكومة يخفض الوكالة.
نظرية السوق الداخلي للأدوات	الأدوات تتنافس وفق تكلفة/خطر/دقة لا وفق التوفر فقط.
نظرية الذاكرة ذات الطبقات	ذاكرة خام، ذاكرة مهارة، ذاكرة اعتراض، ذاكرة نظرية.
نظرية التحقق المزدوج	لا تقبل النتيجة حتى تمر عبر معيار مستقل أو أثر خارجي.

هذا القاموس لا يغلق المعنى؛ بل يضع النسخة التشغيلية الأولى. كل مصطلح لا ينتج أداة أو اختبارا يجب أن يحذف أو يدمج في نسخة لاحقة.

المراجع المختارة

1. V, Arunkumar; Gangadharan, G. R.; Buyya, Rajkumar. *Agentic Artificial Intelligence (AI): Architectures, Taxonomies, and Evaluation of Large Language Model Agents*. arXiv:2601.12560, 2026.
2. Plaat, Aske et al.. *Agentic Large Language Models, a Survey*. arXiv:2503.23037, 2025.
3. Gao, Yunfan et al.. *Retrieval-Augmented Generation for Large Language Models: A Survey*. arXiv:2312.10997, 2023.
4. Peng, Boci et al.. *Graph Retrieval-Augmented Generation: A Survey*. arXiv:2408.08921, 2024.
5. Edge, Darren et al.. *From Local to Global: A Graph RAG Approach to Query-Focused Summarization*. arXiv:2404.16130, 2024.
6. Yao, Shunyu et al.. *ReAct: Synergizing Reasoning and Acting in Language Models*. arXiv:2210.03629, 2022.
7. Schick, Timo et al.. *Toolformer: Language Models Can Teach Themselves to Use Tools*. arXiv:2302.04761, 2023.
8. Packer, Charles et al.. *MemGPT: Towards LLMs as Operating Systems*. arXiv:2310.08560, 2023.
9. Wang, Guanzhi et al.. *Voyager: An Open-Ended Embodied Agent with Large Language Models*. arXiv:2305.16291, 2023.
10. Park, Joon Sung et al.. *Generative Agents: Interactive Simulacra of Human Behavior*. arXiv:2304.03442 / UIST 2023, 2023.
11. Zhou, Shuyan et al.. *WebArena: A Realistic Web Environment for Building Autonomous Agents*. arXiv:2307.13854 / ICLR 2024, 2023/2024.
12. Jimenez, Carlos E. et al.. *SWE-bench: Can Language Models Resolve Real-World GitHub Issues?*. arXiv:2310.06770 / ICLR 2024, 2023/2024.
13. Model Context Protocol Authors. *Model Context Protocol Specification*. modelcontextprotocol.io/specification/2025-11-25, 2025.
14. Tran, K. T. et al.. *Multi-Agent Collaboration Mechanisms: A Survey of LLMs*. arXiv:2501.06322, 2025.
15. Singh, A. et al.. *Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG*. arXiv:2501.09136, 2025.