نواة مساعف لإعادة التدريب المستمر
وإعادة رسم الخرائط المعرفية في الأنظمة الذكية الوكيلية

معمارية حاكمية AGI-Oriented لتحويل كل KB / OS / HyperSkill / HyperLoop
إلى تحديث معرفي قابل للتحقق، التدريب، النشر، والتراجع

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

النطاق الهندسي
شبكة افتراضية قابلة للعنونة حتى **20,000,000 node**، وسعة حاكمية حتى **1,000,000 candidate delta**، مع فصل صارم بين RAG وGraphRAG وSFT وPreference Optimization وLoRA/QLoRA وContinued Pretraining.

ورقة بحثية نظرية-هندسية - نسخة 1.0 - يوليو 2026

الملخص والكلمات المفتاحية

تقترح هذه الورقة معمارية متكاملة لتحويل المعرفة الجديدة من مجرد ملفات نصية أو قواعد معرفة منفصلة إلى تغييرات محكومة داخل نواة نظام تفكير ونظام وكيلي. تنطلق المعمارية من أن إضافة ملف جديد إلى السياق لا تعادل تدريب النموذج، وأن إعادة التدريب العشوائي قد ترفع الكلفة وتسبب النسيان الكارثي والتضارب الدلالي وتوسيع الصلاحيات بلا مسوغ. لذلك تُعرَّف عملية موحدة تبدأ باكتشاف الأصل المعرفي، وحسابه تشفيرياً، وتحديد الحقوق والحدود، ثم تفكيكه إلى وحدات ادعاء وآلية ودليل وحدّ تطبيقي، قبل رسم أثره في شبكة المعرفة واختيار أقل نمط تحديث تدخلاً.

تقدّم الورقة مفهوم **Knowledge-to-Kernel Governance** بوصفه طبقة حاكمية تفصل بين أربعة أشياء كثيراً ما تختلط: استرجاع المعرفة، تعديل خريطة العلاقات، اكتساب سلوك قابل للنقل، وتعديل الأوزان. كما تصوغ تمثيلاً افتراضياً لشبكة تصل إلى عشرين مليون عقدة، مع مليون خانة تغيير محتملة لا تُعد تغييرات مطبقة إلا بعد وجود مصدر فعلي، وحالة قديمة وجديدة، واختبارات، ومراجع مستقل، ومؤشر تراجع. ويُربط ذلك بنظريات مساعف: MCD وQ-MCD وHAKIM-MCD وAGI-Fattah، وبمصانع HyperSkills وHyperLoops والذاكرة ذات النسب المعرفي.

النتيجة الأساسية هي أن "التدريب المستمر" لا ينبغي أن يكون حدثاً واحداً، بل سلسلة قرارات: هل تكفي إضافة RAG؟ هل تحتاج المعرفة إلى GraphRAG remap؟ هل تحولت إلى مهارة متكررة؟ هل توجد أمثلة إيجابية وسلبية وقياسات تمنع التسرب والنسيان؟ وهل أثر التغيير في العالم الخارجي أفضل من خط الأساس؟ بهذا تتحول النواة من مخزن ضخم إلى نظام تطوري قابل للفحص والتراجع.

الكلمات المفتاحية

AGI-Oriented OS; Continuous Post-Training; Knowledge Governance; HyperRouting; HyperSkill; HyperLoop; Graph Remapping; MCD; Q-MCD; Falsification Courts; World-Effect Evaluation; Self-Reconstruction Memory.

القيمة العلمية للملخص

تكمن قيمة المقترح في أنه لا يساوي بين الحفظ والتعلم. فالمعلومة قد تكون متاحة للنموذج عن طريق الاسترجاع، ومع ذلك يظل عاجزاً عن توظيفها ضمن قرار أو أداة أو مسار تحقق. والعكس صحيح: قد يكتسب النموذج سلوكاً عاماً من أمثلة قليلة من دون أن يحمل التفاصيل الواقعية داخل أوزانه. لذا تقيس المعمارية جودة التحويل بين مستويات المعرفة، لا مجرد حجم corpus أو عدد مرات fine-tuning.

- الفصل بين معرفة الشيء ومعرفة كيفية استعماله.
- اعتبار التراجع جزءاً من التعلم لا إجراءً طارئاً.
- ربط كل ترقية بمقياس خارجي ونسخة سابقة محفوظة.

1. المشكلة البحثية وحدود الادعاء

تواجه مشاريع الأنظمة الذكية الحديثة مفارقة واضحة: تزداد ملفات المعرفة والمهارات والأدوات بسرعة، لكن النواة التشغيلية لا تملك دائماً معياراً لمعرفة ما الذي ينبغي استرجاعه، وما الذي ينبغي تحويله إلى سياسة، وما الذي يستحق تدريباً سلوكياً، وما الذي يتطلب تعديلاً فعلياً للأوزان. عند غياب هذا الفصل، تتحول المعرفة إلى تضخم سياقي، وتتكرر المهارات بأسماء مختلفة، وتصبح خرائط الوكلاء غير مستقرة، ويصعب معرفة سبب نجاح القرار أو فشله.

المشكلة ليست حجم البيانات وحده، بل غياب **حاكمية الاختلاف الفعّال**. قد يحتوي ملف من ألف صفحة على تغيير واحد يبدل قرار النظام، وقد يحتوي ملف صغير على قاعدة دستورية عالية الأثر. لذلك لا يجوز أن يكون الحجم أو الجدة اللغوية مؤشراً للترقية. يجب تحديد الفرق الذي يغيّر تعريفاً أو آلية أو مساراً أو صلاحية أو اختباراً أو أثراً خارجياً، ثم قياس اتساعه ومخاطره وقابليته للنقل.

تلتزم الورقة بحد إبستمولوجي صريح: المعمارية AGI-oriented وليست برهاناً على تحقق AGI. وملف Markdown لا يعيد تدريب الأوزان الخفية بذاته. التدريب الفعلي يحتاج نموذجاً يمكن الوصول إلى نقاطه المرجعية، وبيانات مرخّصة، وإطار تدريب، وحوسبة، وسجلات تجارب، ومجموعات اختبار غير ملوثة. ما تقدمه الورقة هو نظام قرار وتنظيم وتحويل واختبار يجعل تلك العملية قابلة للتكرار والمراجعة.

سؤال البحث المركزي

كيف يمكن تحويل كل معرفة جديدة إلى **delta** محدد الأثر، ثم توجيهه إلى أقل نمط تحديث كلفةً وأعلى نمط تحققاً، مع حماية الدستور والذاكرة والصلاحيات ومنع النسيان والتضارب؟

تفكيك المشكلة إلى طبقات

يمكن رد معظم إخفاقات أنظمة الدمج إلى واحد من ستة أسباب: أصل معرفي غير موثوق، تكرار غير مكتشف، خريطة علاقات ناقصة، سلوك غير متعلم، أداة أو صلاحية غير مناسبة، أو تقييم يقيس مؤشراً بديلاً عن النجاح الحقيقي. وتسمح هذه القسمة بتجنب العلاج المفرط؛ فالمشكلة في الفهرس لا تحتاج نموذجاً أكبر، والمشكلة في السياسة لا يحلها تضخيم قاعدة المعرفة.

- تشخيص سبب الفشل قبل اختيار التقنية.
- منع استعمال التدريب كحل افتراضي لكل نقص.
- تسجيل ما لم يتغير بقدر تسجيل ما تغير.

2. الفرضيات والأسس النظرية

تبنى المعمارية على خمس فرضيات. الأولى أن الذكاء التشغيلي لا يساوي كمية النص المحمّل، بل جودة الفصل بين السؤال والاختلاف والدليل والآلية والفعل. الثانية أن المعرفة المتغيرة زمنياً أو الواقعية غالباً ما تُدار أفضل عبر الاسترجاع، بينما السلوكيات المستقرة المتكررة قد تستحق التدريب. الثالثة أن العلاقات بين الوحدات ليست هامشية؛ إذ قد لا تتغير العقدة نفسها لكن تتغير روابط الاعتماد والتعارض والترتيب، وهنا تظهر قيمة إعادة رسم الخريطة.

الفرضية الرابعة أن كل قدرة وكيلية يجب أن تُربط بصلاحيات وحدود وميزانية وتوقف وتراجع. فالوكيل غير المحدود ليس أكثر ذكاءً، بل أقل قابلية للضبط. والفرضية الخامسة أن القياس الداخلي للنموذج غير كافٍ: النجاح يُقاس في **World Effect**، أي أثر قابل للملاحظة أو التنفيذ أو التحقق خارج السرد الذاتي للنظام.

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

تستعمل الورقة معادلة MCD كدالة إرشادية: جودة التمييز D، التكامل I، قابلية النقل T، الوكالة المحدودة A، القوة التنبؤية أو العملية P، النقد والتصحيح C، مطروحاً منها الضجيج N. ليست هذه معادلة فيزيائية مكتملة، بل إطار وزن يساعد على منع ترقية المعرفة لمجرد كثافتها البلاغية.

مبدأ الحاكمية

كل تحديث كبير يجب أن يكون قابلاً للتحليل إلى: **سبب، فرق، مسار أثر، اختبار، صلاحية، إصدار، ومؤشر تراجع.**

صلة الفرضيات بالهندسة

تتحول الفرضيات النظرية إلى ضوابط قابلة للبرمجة: يُقاس الضجيج بنسبة الوحدات غير المستخدمة، وتُقاس قابلية النقل بأداء مهام لم تظهر أثناء التدريب، وتُقاس الوكالة المحدودة بصحة قرارات التوقف والتفويض. بهذا لا تبقى MCD وصفاً بل تصبح لوحة مؤشرات تساعد الفريق على تفسير سبب ترقية أصل أو رفضه، مع الإقرار بأن الأوزان تحتاج معايرة تجريبية.

- استخدام مؤشرات متعددة بدل رقم سحري واحد.
- معايرة الأوزان حسب نوع المهمة والخطر.
- حفظ تبرير القرار مع النتيجة التجريبية.

3. النواة الدستورية وحماية الهوية

النواة الدستورية هي الطبقة التي تحدد ما لا يجوز للأصول العادية تغييره تلقائياً: هوية النظام، أولوية الحقيقة على الطلاقة، حدود الصلاحيات، قواعد الخصوصية، لزوم المصدر، ومنع الادعاء غير المدعوم. وعليه، لا تُمنح قاعدة معرفة صفة دستورية لأنها تحمل كلمة OS أو "حاكمية". يجب أن تمر التعديلات الدستورية بمسار مستقل يتضمن موافقة بشرية صريحة، وتحليل تناقض، ونسباً بين القاعدة القديمة والجديدة، واختبارات رجعية، وخطة هجرة وتراجع.

تُطبق آلية **fail-closed**: إذا غاب التفويض أو المراجع أو شروط التوقف أو مسار التراجع، يُرفض الانتقال إلى فعل مؤثر. ولا يعني "الإجبار" تجاوز سياسات النموذج أو المنصة؛ بل يعني إجبار النظام نفسه على احترام شروطه قبل التنفيذ. هذه إعادة تعريف ضرورية لأن استعمال كلمة forced في بيئة وكيلية بلا قيود قد يُفهم كتوسيع غير مشروع للصلاحيات.

الحالة	القرار الحاكم
مصدر غير معروف	عزل الأصل وعدم دمجه
حقوق استعمال غير محسومة	استرجاع داخلي محدود أو رفض التدريب
تعارض مع الدستور	إحالة إلى مراجعة بشرية مستقلة
فائدة بلا أثر قابل للقياس	إبقاء الأصل في طبقة الاسترجاع
تحسن مع نسيان عام مرتفع	رفض الترقية أو عزل Adapter
غياب التراجع	منع النشر الإنتاجي

بهذا تصبح الهوية ليست فقرة افتتاحية، بل طبقة تنفيذية تقطع مسار النشر عندما تتغير شروط الثقة.

سيناريو دستوري

إذا وصل OS جديد يطلب منح الوكلاء صلاحية نشر تلقائي، لا يُعامل الطلب بوصفه معرفة تقنية فقط. تفصل النواة بين الجزء المفيد، مثل قالب إعداد المحتوى، وبين الجزء السيادي الذي يوسّع الصلاحيات. يمكن استيعاب القالب كمهارة، بينما يبقى النشر خلف بوابة موافقة. هذا الفصل يسمح بالاستفادة من الأصل دون السماح له بإعادة تعريف السلطة من داخل نفسه.

- فصل المحتوى عن سلطة التعليمات.
- اعتماد مسار خاص للتغييرات السيادية.
- إبقاء الإنسان طرفاً في الأفعال العالية الأثر.

4. بروتوكول قبول المعرفة الجديدة

يمر كل أصل جديد عبر سجل قبول موحد. تبدأ العملية بحساب content hash وإنشاء هوية ثابتة للأصل، ثم تسجيل المالك والمسار والتاريخ والحقوق والحساسية والمشروع الذي ينتمي إليه. بعد ذلك يُفكك النص إلى وحدات: تعريفات، دعاوى، آليات، أدلة، أمثلة، أدوات، اختبارات، حدود، ومقترحات. هذا التفكيك يمنع نقل الملف كله إلى النواة عندما تكون القيمة مركزة في وحدات قليلة.

تُصنّف كل دعوى ضمن درجات: حقيقة مثبتة، إرشاد موثق، تفسير، تركيب نظري، فرضية، اقتراح، استعارة، أو تخمين. كما يُسجل تاريخ المصدر لأن بعض المعرفة البرمجية والسوقية والتنظيمية تتقادم سريعاً. ثم يُنفّذ كشف التكرار الدلالي والتعارض، ولا يُحذف التعارض تلقائياً، بل يُحفظ مع روابط "يناقض"، "يخصّص"، "ينسخ"، أو "أحدث من".

- إثبات التفويض والحقوق قبل أي تدريب أو نشر.
- استخراج Active Difference الذي يغير قراراً أو قدرة أو خريطة.
- تحديد الطبقات والمهارات والحلقات والاختبارات المتأثرة.
- اختيار نمط التحديث الأقل تدخلاً.
- إنشاء مجموعة تحقق وتراجع قبل الترقية.
- حفظ نسب الأصل وإصداراته وعدم الكتابة الصامتة فوق الذاكرة.

قاعدة الكفاية

إذا أمكن تحقيق المنفعة باسترجاع موثوق، فلا يُنقل التغيير إلى الأوزان لمجرد الرغبة في "إعادة التدريب".

مثال قبول عملي

عند إدخال بحث جديد عن GraphRAG، ينشئ النظام بطاقة مصدر، ويستخرج التعريفات والآليات والنتائج والحدود، ثم يقارنها بما هو موجود. فإذا كانت الجدة مجرد صياغة لا يضاف شيء. وإذا ظهرت آلية استرجاع مختلفة، تُنشأ دلتا للعلاقة بين query وsubgraph مع اختبار استدعاء. ولا تتحول الورقة بكاملها إلى SFT إلا إذا احتوت سلوكاً ثابتاً يمكن تقييمه.

- تجنب نسخ الورقة كلها إلى بيانات التدريب.
- تحويل الحدود المذكورة في البحث إلى حالات اختبار.
- إبقاء النتائج التجريبية مرتبطة بسياقها الأصلي.

5. استخراج الاختلاف الفعّال

الاختلاف الفعّال هو أصغر وحدة تغيير قادرة على تعديل نتيجة مهمة. يمكن أن يكون تعريفاً أكثر دقة، أو شرطاً جديداً، أو علاقة سببية، أو اختباراً يكشف فشلاً كان مخفياً، أو صلاحية تُمنع، أو أداةً تغير مسار التنفيذ. وبدلاً من سؤال "ماذا يقول الملف؟" تسأل النواة: "ما الذي سيتغير لو قبلناه؟ وما الذي لا ينبغي أن يتغير؟".

يُكتب لكل اختلاف سجل يتضمن الحالة القديمة، والحالة المقترحة، والثقة، والأدلة، والاعتمادات، والأثر المحلي والمتعدي، والاختبارات، وحد التراجع. هذا السجل هو الجسر بين المعرفة والتدريب؛ فلا تتحول الفقرة إلى مثال تدريبي قبل أن نعرف السلوك المراد اكتسابه والسلوك المرفوض والقياس الذي يثبت التحسن.

$$\Delta K = NewState - OldState; Apply(\Delta K) \text{ only if } Evidence \wedge Authority \wedge Tests \wedge Rollback$$

نوع الفرق	مثال	نمط التحديث المرجح
واقعي متغير	معلومة إصدار أو قانون حديث	RAG
علاقة جديدة	اعتماد مهارة على أداة	GraphRAG remap
إجراء متكرر	بروتوكول تحقق قابل للنقل	HyperSkill / SFT
ترتيب تفضيل	رفض جواب بلا دليل	Preference Optimization
اختصاص عميق ثابت	لغة أو مجال واسع مرخص	LoRA أو Continued Pretraining

وبذلك تُقاوم المعمارية وهم أن كل جديد يجب أن يدخل "النواة الداخلية" بالمعنى نفسه.

بطاقة الاختلاف

يمكن تمثيل الفرق في بطاقة قصيرة تقرؤها الآلات والبشر: ما الدعوى القديمة؟ ما الجديد؟ ما سبب الثقة؟ ما العقد التي نتأثر؟ ما الاختبار الذي يفشل قبل التغيير وينجح بعده؟ وما شرط إلغاء التغيير؟ هذه البطاقة تمنع أن تتحول عملية التطوير إلى تراكم commits غير مفهوم، وتجعل المقارنة بين تحديثات متنافسة ممكنة.

- Old state وNew state صريحان.
- اختبار يميز النسختين لا اختبار شكلي.
- Rollback pointer مرتبط بالإصدار السابق.

6. فضاء العشرين مليون عقدة

تقترح الورقة فضاءً افتراضياً قابلاً للعنونة من MAGI-NODE-00000001 إلى MAGI-NODE-20000000. كلمة "افتراضي" جوهرية: لا يُشترط إنشاء عشرين مليون سجل في ملف واحد، بل تُستعمل النطاقات والشظايا والأسماء المعيارية والعناوين المبنية على المحتوى لتمثيل قابلية التوسع. عند توفر قاعدة رسوم بيانية خارجية يمكن مادية العقد تدريجياً حسب الحاجة.

تنقسم العقد إلى عائلات: مفاهيم، دعاوى، مصادر، آليات، اختبارات، مهارات، حلقات، وكلاء، أدوات، صلاحيات، مشاريع، نماذج، ذاكرات، ونتائج عالمية. وتُكتب الحواف بنوع واضح مثل "يعتمد على"، "يتحقق بواسطة"، "يتعارض مع"، "يولّد"، "يستدعي"، "يُحظر بواسطة"، و"يُستبدل بـ". هذا يمنع الخريطة من أن تكون مجرد تشابه متجهي غير قابل للتفسير.

يُستخدم الاسترجاع المتناثر بدل تحميل الشبكة كاملة. لكل مهمة تُبنى **حزمة سياق دنيا** تحتوي الدستور المختصر، ووحدات المعرفة الأقرب، والمهارات اللازمة، والحلقات المرتبطة، والاختبارات الحرجة. وبهذا يصبح الاتساع ميزة تنظيمية لا عبئاً رمزياً.

> **مبدأ العنونة لا الادعاء**
>
> القدرة على عنونة 20 مليون عقدة لا تعني أن 20 مليون معرفة صحيحة أو مدرَّبة موجودة فعلاً؛ إنها سعة هندسية يجب ملؤها بأصول موثقة.

> **التقسيم إلى شظايا**
>
> ينبغي تقسيم فضاء العقد إلى namespaces بحسب المشروع والمجال ونوع الذاكرة، مع معرّفات مشتركة للكيانات العابرة. ويمكن تخزين الحواف في ملفات متناثرة أو قاعدة رسوم بيانية، بينما تحتفظ ملفات Markdown بالوصف الإنساني والعقود. هذا التصميم الهجين يمنع حبس النظام في صيغة واحدة، ويجعل نقله بين Hermes أو بيئة محلية أو cluster تدريب أكثر سهولة.
>
> - معرّفات ثابتة عبر البيئات.
> - فصل الوصف البشري عن فهرس التنفيذ.
> - تركيب شظايا السياق عند الطلب فقط.

7. مليون خانة تغيير وإعادة الرسم

لتمثيل مليون تغيير محتمل، تقترح المعمارية خمسين ألف خلية إعادة رسم، في كل خلية عشرون خانة delta. الخانة ليست تغييراً منجزاً، إنها عقد حاكمية ينتظر أصلاً فعلياً. ولا تُعد مطبقة حتى تضم معرّف المصدر، والتجزئة، والحالة القديمة والجديدة، والعقد والحواف المتأثرة، والاختبارات، والمراجع، والحكم، ومؤشر التراجع.

تفيد هذه البنية في فصل السعة النظرية عن الإنجاز الواقعي. فهي تسمح بتخطيط مليون تعديل من دون اختلاق محتوى أو ادعاء تدريب لم يحدث. كما تجعل كل موجة تحديث قابلة للتجزئة إلى شظايا، فلا ينتشر التغيير دفعة واحدة في النظام. يحدَّد مخروط الأثر بعمق أقصى وثقة ومخاطر وقيمة معلوماتية.

$$\text{Impact}(\delta) = \text{LocalChange} \times \text{Confidence} \times \text{EdgeCriticality} \times \text{Depth} \times \text{Reversibility}$$

إذا غيّر delta تعريف أداة مثلاً، تُستخرج الحلقات التي تستدعيها، والمهارات التي تفترض مخرجاتها، والاختبارات التي تعتمد عليها، والسياسات التي تحد صلاحيتها. تبطل الذاكرات المؤقتة القديمة، وتُعاد فهرسة الأجزاء المتأثرة فقط، ثم تُعاد المهام الحرجة على الخريطة القديمة والجديدة.

شرط التطبيق

لا انتشار بلا **impact cone** محدد، ولا ترقية شظية بلا مقارنة قبل/بعد، ولا مسح للإصدار السابق قبل مرور فترة مراقبة.

حوكمة التغيير على دفعات

يفضل تجميع الدلتا في releases صغيرة ذات موضوع واحد، لا في دفعة مليونية متجانسة. يتيح ذلك اختبار الأثر السببي، لأن تغير مئات السياسات والنماذج في وقت واحد يجعل فشل النتيجة غير قابل للتشخيص. وتستطيع الخلايا العشرين داخل كل remap cell تمثيل ذرّات مترابطة، مع ترقية بعضها وحجر بعضها الآخر بدلاً من الحكم على الحزمة كلها.

- إصدارات صغيرة قابلة للعزل.
- توقيع manifest لكل شظية.
- فترة مراقبة قبل حذف النسخ السابقة.

8. موجّه أنماط التحديث

القرار الأهم ليس "هل نضيف المعرفة؟" بل "بأي طبقة؟". يرتب الموجّه الأنماط من الأقل تدخلاً إلى الأعلى. يبدأ بتعديل موجّه أو بروتوكول قصير، ثم تركيب استرجاعي، ثم إعادة رسم بياني، ثم تسجيل مهارة أو حلقة، ثم تحديث موجّه الاختيار، ثم التدريب السلوكي والتفضيلي، ثم المحولات منخفضة الرتبة، وأخيراً التدريب المستمر أو الكامل.

النمط	متى يُختار	الخطر المركزي
Prompt Kernel	قاعدة إجرائية قصيرة وثابتة	تضخم التعليمات
RAG	حقائق كثيرة أو متغيرة	سوء الاسترجاع
GraphRAG	علاقات واعتمادات معقدة	عدم اتساق الحواف
HyperSkill	سلوك قابل لإعادة الاستخدام	ترقية مبكرة
HyperLoop	تنفيذ تكراري محدود	حلقة غير منتهية
SFT	أمثلة ثابتة عالية الجودة	فرط التخصيص
Preference	تعلم الاختيار والرفض	تحيز المقارنات
LoRA/QLoRA	اختصاص سلوكي معزول	تداخل المحولات
Continued Pretraining	جسم معرفي واسع متجانس	نسيان وكلفة وحقوق

القاعدة هي **Minimum-Invasive Adequacy**: اختر أقل تغيير يحقق الأثر المطلوب ويحافظ على القابلية للتراجع. فإذا كانت المشكلة ضعف استرجاع فلا تُعالج بتعديل الأوزان، وإذا كانت المشكلة سلوكاً متكرراً لا يكفي أن يبقى في وثيقة طويلة غير قابلة للتنفيذ.

دالة قرار مختصرة

$$\text{TrainValue} = \text{TransferGain} + \text{ToolGain} + \text{CalibrationGain} - \text{ForgettingRisk} - \text{SecurityRisk} - \text{MaintenanceCost}$$

الموجه بوصفه سياسة تعلم

الموجّه نفسه يحتاج إلى بيانات تدريب: حالات نجح فيها RAG وحالات احتاجت Adapter، وحالات تسبب فيها التدريب غير الضروري في نسيان أو كلفة. وبدلاً من قواعد ثابتة فقط، يمكن بناء classifier أو policy model يقترح النمط، على أن يظل القرار العالي الأثر قابلاً للتفسير والمراجعة. وبذلك يصبح اختيار طريقة التعلم مهارة تتطور من النتائج المتراكمة.

- تدريب الموجه على outcomes لا على التسميات فقط.
- إظهار أسباب اختيار النمط.
- عدم السماح للموجه بتجاوز الدستور.

9. تدريب نموذج التفكير

يُدرَّب نموذج التفكير في هذه المعمارية على مخرجات قابلة للفحص، لا على مطالبةٍ بإظهار السلسلة الداخلية الخاصة. تشمل وحدة التدريب: التعليمة، وحزمة الدليل، وبيان الفرق الفعّال، وتعليلاً موجزاً، وخطة، وقرار صلاحية، وأثر أداة أو محاكاة جافة، ونقداً، وجواباً، وسلوكاً مرفوضاً، وبقية معرفية قابلة للنقل. هذا يحقق فائدة الاستدلال المنظم من دون تحويل البيانات إلى طلب دائم لكشف أفكار داخلية.

تُصمم الأمثلة السلبية بعناية: جواب طليق بلا دليل، خلط بين اقتراح وحقيقة، حلقة غير محدودة، فعل بلا تفويض، استشهاد مختلق، تقييم ذاتي بلا أثر خارجي، أو إعلان نجاح التدريب بلا نقطة تحقق وسجل تشغيل. وتمثل هذه السلبيات "محكمة" مضمنة في البيانات بدلاً من الاقتصار على أجوبة مثالية.

يحتاج التدريب إلى تقسيم زمني وموضوعي ومشروعي. يجب ألا تتسرب نسخ الوثائق نفسها إلى الاختبار، كما ينبغي وجود مهام لا تشبه الصياغة التدريبية لكنها تقيس الآلية نفسها. يُقاس التحسن في الدقة والتعاير واختيار الأداة ونقل السلوك، مع اختبار أن النموذج لم يفقد قدراته العامة.

سجل التدريب القياسي

instruction → evidence → active difference → concise rationale → plan → permission → tool/dry-run → verifier → answer → rejected behavior → memory residue

تنسيق البيانات

يمكن تصدير وحدات التفكير إلى JSONL أو صيغة محادثة، لكن يحتفظ السجل الأصلي بعلاقات المصدر والاختبار. ويفضل ألا تكون كل الأمثلة طويلة؛ فالتنوع بين قرارات قصيرة ومسارات مركبة يمنع النموذج من ربط الجودة بالاستطراد. كما يجب تدريب النموذج على قول لا أعلم أو أحتاج أداة عندما تكون الأدلة ناقصة.

- أمثلة قصيرة وطويلة ومتعددة المجالات.
- تمييز rationale موجز عن النص النهائي.
- تضمين الامتناع والتصعيد كسلوكين صحيحين.

10. التدريب الوكيلي وما بعد التدريب

النموذج الوكيلي لا يُقاس بقدرته على كتابة خطة جميلة، بل بقدرته على تنفيذ مسار محدود مع صلاحيات واضحة واستعادة صحيحة عند الفشل. لذلك يرتبط كل مثال تدريبي بعقدة فعل أو محاكاة فعل، ويُسجل استدعاء الأداة ومدخلاته ومخرجاته وأخطاؤه وإعادات المحاولة وكلفته، مع حذف الأسرار والبيانات غير الضرورية.

يتضمن ما بعد التدريب مهام اختيار الأداة، ورفض الأداة غير المصرح بها، والتوقف عند غياب شرط، والعودة إلى نقطة تحقق، وطلب موافقة بشرية عند الأثر العالي. يمكن استخدام SFT لتعليم البنية الأساسية، ثم Preference Optimization لترجيح السلوك الأكثر تحققاً وأقل مخاطرة، مع إبقاء اختبارات مستقلة للأمان لأن تفضيلات البيانات قد تكون ناقصة أو متحيزة.

- ميزانية محاولات ورموز وزمن وكلفة لكل مهمة.
- حالة واضحة: قراءة فقط، محاكاة، تجهيز، أو تنفيذ مصرح.
- فصل المخطط والمنفذ والمراجع في المهام المتوسطة والعالية الأثر.
- سجل كامل للقرار لا يكشف مفاتيح أو أسراراً.
- اختبار العودة بعد الخطأ وليس النجاح فقط.
- تحويل النتيجة إلى ذاكرة بعد تحقق خارجي.

معيار الوكالة

الوكالة الجيدة = **قرار صحيح + صلاحية صحيحة + أداة صحيحة + توقف صحيح + أثر صحيح + ذاكرة صحيحة.**

من المختبر إلى الفعل

لتحويل التدريب الوكيلي إلى قدرة واقعية، يجب أن تحتوي بيئة الاختبار على أدوات أو محاكيات بواجهات ثابتة وأخطاء مقصودة. فالنموذج الذي ينجح عندما تعمل الأداة دائماً لا يكون قد تعلم التعافي. تُحقن حالات timeout وpermission denied ونتائج جزئية، ويُقاس هل يكرر بأمان أو يغير الخطة أو يتوقف ويطلب تدخلاً.

- اختبار الأخطاء المتوقعة وغير المتوقعة.
- قياس صحة retries لا عددها.
- منع إعادة المحاولة بعد فشل تفويض صريح.

11. مصنع HyperSkills

الـ HyperSkill ليست فقرة معلوماتية، بل عقد قدرة قابلة للاستدعاء. تتكون من بصمة نية، ومدخلات ومخرجات نوعية، ووكيل مالك، وأدوات وصلاحيات، وحلقة افتراضية، ومراجع، واختبارات، وقواعد ترقية وإهمال. لا تُرقّى القدرة من نجاح واحد؛ تقترح المعمارية ثلاثة انتصارات موثقة في ثلاث عائلات مهام على الأقل، مع غياب فشل حرج غير محلول.

تُستخرج المهارات من المعرفة الجديدة عندما يظهر نمط متكرر يغير العمل. مثال ذلك: "تحويل ورقة بحثية إلى سجل ادعاءات مع درجات دليل واختبارات نقض". أما معلومة منفردة عن إصدار مكتبة فلا تصبح مهارة، بل تبقى في الاسترجاع. هذا الفصل يمنع انفجار سجل المهارات إلى أسماء متشابهة بلا قيمة تنفيذية.

حقل المهارة	مضمونه
intent_signature	متى تُستدعى وما الذي لا تغطيه
input/output contract	أنواع البيانات والقيود
owner/verifier	المسؤول ومن يراجع
tool/permission profile	الأدوات والحد الأدنى من الصلاحية
benchmarks	اختبارات الهدف والنقل والفشل
promotion/deprecation	شروط الترقية والإيقاف

تُربط المهارات بالعقد المفاهيمية والمشاريع والنماذج المناسبة عبر HyperRouting متناثر، فلا تُحمّل جميع المهارات في كل مهمة.

دورة حياة المهارة

تمر المهارة بمراحل candidate وstaged وapproved وdeprecated. ويجب أن تحمل changelog واختبارات إصدار، لأن تغير أداة خارجية قد يبطل العقد حتى لو بقي وصف المهارة صحيحاً. وعند دمج مهارتين متشابهتين، تحفظ aliases وروابط الترحيل حتى لا تنكسر الحلقات القديمة. هذه الإدارة تجعل سجل القدرات أقرب إلى package registry منه إلى قائمة prompts.

- نسخ دلالية للمهارات.
- اختبارات توافق مع الأدوات.
- خطة ترحيل عند الدمج أو إعادة التسمية.

12. مصنع HyperLoops

الـ HyperLoop هو برنامج تكراري محدود وليس شعاراً للاستمرار. يتضمن محفزاً، وحالة، وميزانيات، وشروط توقف، وبوابات موافقة، ونقاط تحقق، ومراجعاً، وتراجعاً، وحقول أثر عالمي. يدور المسار: ملاحظة، سؤال، تمييز، استرجاع، تخطيط، تفويض، فعل، تحقق، إصلاح، ثم ذاكرة.

يُحظر تصميم حلقة غرضها "الاستمرار حتى النجاح" بلا حد؛ لأن ذلك يحول عدم اليقين إلى استنزاف أو سلوك خطر. بدلاً من ذلك تُحدد حالات توقف النجاح، وعدم التقدم، واستنفاد الميزانية، وغياب التفويض، ورفض المراجع، وارتفاع الخطر. وقد يكون أفضل مخرج للحلقة تقريراً يبين أن شروط التنفيذ لم تكتمل.

Loop = Observe → Question → Differentiate → Retrieve → Plan → Authorize → Act → Verify → Repair → Remember

عند تغيير أصل معرفي، تحدد خريطة الأثر الحلقات التي تعتمد على افتراض تغير. تُعاد محاكاتها على حالات سابقة وحالات خصمية، وتُقارن معدلات التوقف الصحيح والتعافي والتكلفة. وبهذا يصبح إعادة الرسم مرتبطة فعلاً بالتنفيذ لا بمجرد روابط نصية.

شرط الترقية

لا تُرقّى حلقة قبل نجاح اختبارات **التوقف، الإصلاح، التراجع، وWorld Effect.**

حلقات متعددة المقاييس

ليست كل الحلقات من المستوى نفسه. توجد micro-loop للتحقق من خطوة، وtask-loop لإكمال مهمة، وproject-loop لمراجعة سلسلة مخرجات، وlearning-loop لترقية مهارة بعد نتائج متكررة. يمنع الفصل أن تستعمل حلقة التعلم البطيئة داخل كل استدعاء أو أن تُترجم نتيجة مهمة واحدة مباشرة إلى تغيير دائم.

- Micro-loop سريع ومحلي.
- Task-loop محدود بالهدف والميزانية.
- Learning-loop بطيء ويحتاج أدلة تراكمية.

13. الذاكرة ذات النسب المعرفي

تقسم المعمارية الذاكرة إلى دلالية، وحدثية، وإجرائية، وهووية. تخزن الدلالية المفاهيم والدعاوى المستقرة، وتخزن الحدثية مسارات المهام ونتائجها، وتخزن الإجرائية المهارات والحلقات المجربة، بينما تحفظ الهووية القواعد الدستورية وحدود المشاريع. هذه الأقسام لا تُدمج في كتلة واحدة لأن لكل منها سياسة تحديث وتقادم مختلفة.

تُحظر الكتابة الصامتة فوق الذاكرة. كل تعديل يحفظ الحالة السابقة والمقترحة والدليل والثقة والحكم والإصدار وروابط الاستبدال. وتبقى الذاكرة المقترحة في حجر صحي حتى تمر بالاختبار. وعند وجود روايتين متعارضتين، لا تختزل إحداهما تلقائياً، بل تُربطان ويُوجه الاسترجاع إلى الأحدث أو الأقوى حسب المهمة.

نوع الذاكرة	سياسة التحديث
Semantic	مصدر وثقة وتاريخ وتعريف واضح
Episodic	نتيجة واقعية وفشل وسياق مهمة
Procedural	انتصارات متعددة واختبارات نقل
Identity	موافقة بشرية ومسار دستوري خاص
Proposed	عزل حتى تحقق مستقل

تسمى عملية إدخال النتيجة المحققة **Self-Reconstruction Memory**: لا يتغير النظام لأنه قال إنه تعلم، بل لأنه راجع أثراً خارجياً وربطه بسبب وتحديث محدود.

> **التقادم والنسيان المتعمد**
>
> ليست المحافظة على كل ذاكرة فضيلة. تحتاج النواة إلى نسيان محكوم: إزالة الكاش القديم، وخفض أولوية مصادر متقادمة، وإهمال مهارة لم تعد أدواتها متاحة، مع الاحتفاظ بسجل أرشيفي للنسب. الفرق بين النسيان الكارثي والنسيان الصحي هو أن الثاني مقصود، موثق، وقابل للاستعادة، بينما الأول خسارة غير مقاسة بسبب التدريب.
>
> - TTL للحقائق سريعة التغير.
> - أرشفة بدلاً من المحو النهائي.
> - اختبارات تكشف فقدان المعرفة العامة غير المقصود.

14. Q-MCD وحاكمية السؤال

لا تبدأ جودة النظام من الجواب، بل من السؤال الذي يولد مسار العمل. يضيف Q-MCD دالة حاكمية للسؤال تقيس عمقه ورافعيته وقابليته للتكذيب وعلاقته بالفعل وحدوده وضجيجه. السؤال الرديء قد يدفع أكبر شبكة إلى استرجاع كثيف بلا أثر؛ والسؤال الجيد قد يحدد عقدتين وحافة واحدة يجب تغييرها.

$$\text{Q-MCD} = G(Q) \times [\lambda_1 Dq + \lambda_2 Iq + \lambda_3 Tq + \lambda_4 Aq + \lambda_5 Pq + \lambda_6 Cq - \lambda_7 Nq]$$

تعمل سلسلة الحاكمية على النحو الآتي: سؤال، فرق، درجة MCD، اختيار مسار، وكالة، أثر عالمي، ثم سؤال أعلى. وفي سياق إعادة التدريب، يكون السؤال الحاسم: هل الفشل سببه نقص معرفة، أم علاقة خاطئة، أم سلوك غير متعلم، أم أداة غير مناسبة، أم صلاحية ناقصة، أم مقياس مضلل؟ كل تشخيص يقود إلى نمط تحديث مختلف.

تساعد حاكمية السؤال أيضاً على منع توسع التغيير. إذا كان الهدف تحسين استدعاء وثيقة محددة فلا مبرر لإعادة تدريب النموذج على ملايين الأمثلة. وإذا كان السلوك يفشل عبر مجالات متعددة رغم استرجاع صحيح، يصبح التدريب السلوكي مبرراً أكثر. هذه العلاقة بين التشخيص والتدخل هي أساس الاقتصاد المعرفي.

قانون السؤال الوعائي

السؤال يفتح الوعاء الحسابي المناسب؛ ولا تُستدعى طبقة أو مهارة أو نموذج إلا إذا غيّر استدعاؤها القرار المتوقع.

السؤال كأداة ضغط سياقي

عندما يصاغ السؤال بصيغة قرار، يمكن للنظام حذف كثير من السياق غير المؤثر. فسؤال: ما المعلومات عن هذا المجال؟ يستدعي موسوعة، أما سؤال: ما العقدة التي تفسر فشل هذا الاختبار؟ فيستدعي مساراً ضيقاً. لذلك لا تكون Q-MCD طبقة فلسفية فقط، بل آلية ضغط تقلل الرموز والكلفة وتحسن إمكانية تتبع السبب.

- تحويل الطلب الفضفاض إلى قرار قابل للاختبار.
- ربط كل سؤال بحد توقف.
- توليد سؤال أعلى بعد ملاحظة الأثر.

15. خوارزمية انتشار الأثر

عند قبول delta لا يُنشر مباشرة. تحسب خوارزمية الانتشار الأثر المحلي، ثم الحواف الحرجة، ثم العمق المسموح، ثم قابلية الرجوع. يبدأ التحليل من العقدة المتغيرة ويبحث عن التعريفات والمهارات والحلقات والاختبارات والمشاريع التي تعتمد عليها. تُعطى الأولوية للعلاقات الصريحة على التشابه المتجهي وحده.

يُنتج التحليل ثلاثة مخرجات: قائمة يجب إعادة فهرستها، وقائمة يجب إعادة اختبارها، وقائمة يجب منعها أو عزلها حتى مراجعة بشرية. يمكن أن يكون أثر التغيير صفرياً خارج مشروع واحد، أو واسعاً إذا مسّ دستوراً أو أداة مشتركة. لهذا لا يكفي عدّ الحواف؛ يجب وزن حرجيتها ونوعها وثقة مصدرها.

- إبطال الكاش والفهارس القديمة في الشظية المتأثرة.
- إعادة توليد خرائط الاسترجاع والروابط النوعية.
- اكتشاف المهارات والحلقات ذات الافتراضات القديمة.
- اختيار أصغر مجموعة اختبارات رجعية كافية.
- مقارنة مسارات النموذج القديم والجديد على نفس الحالات.
- حفظ خريطة قبل/بعد مع توقيع الإصدار.

مخروط الأثر

يُحدّ الانتشار بأربعة قيود: **العمق، الثقة، الخطر، وقيمة المعلومات.** ولا تتحول "إعادة الرسم" إلى انتشار شامل غير مبرر.

الانتشار المحافظ

في الأنظمة الحرجة يمكن استعمال نشر محافظ يبدأ بالحواف ذات الثقة العالية فقط، ثم يوسع المخروط بعد نجاح اختبارات محلية. أما الحواف المستنتجة إحصائياً فتبقى بدرجة أقل ولا تُستخدم لمنح صلاحية أو إلغاء اختبار. ويجب تسجيل سبب كل edge حتى لا تتحول الخريطة إلى شبكة كثيفة يتعذر تفسيرها.

- تمييز الحواف الصريحة والمستنتجة.
- حد أقصى لعمق الانتشار.
- مراجعة بشرية للحواف السيادية.

16. محاكم النقض والتدمير

تُنشئ المعمارية مجموعة محاكم لاختبار التحديث من زوايا مختلفة. محكمة التعريف تمنع المصطلحات الغائمة، ومحكمة الحدود تسأل أين لا يعمل الحل، ومحكمة الدليل تراجع الرتبة والتاريخ والمصدر، ومحكمة الآلية تطلب الوسطاء السببيين، ومحكمة البيانات تبحث عن التكرار والتسرب والحقوق، ومحكمة النسيان تختبر القدرات العامة.

تضاف محكمة الأمان والصلاحيات، ومحكمة استعمال الأدوات، ومحكمة الأثر العالمي، ومحكمة مقاومة الحشو، ومحكمة الصيانة والتراجع. الهدف ليس إبطاء كل تحديث، بل منع انتقال تحديث براق إلى الإنتاج قبل اكتشاف العيوب التي لا تظهر في اختبار الهدف الضيق.

المحكمة	السؤال القاطع
Evidence	هل الادعاء موثق أم مولد؟
Mechanism	كيف ينتج الأثر وتحت أي شروط؟
Duplicate/Conflict	هل يكرر أو يناقض ما هو قائم؟
Forgetting	ماذا خسر النموذج مقابل ما كسب؟
Security	هل وسّع التحديث الصلاحيات أو سطح الهجوم؟
World Effect	هل تغير شيء قابل للتحقق خارج السرد؟
Rollback	هل يمكن العودة كاملة وبسرعة؟

تصدر المحكمة حكماً من ست حالات: ترقية، ترقية محدودة، استمرار في التجهيز، حجر، إهمال، أو تراجع.

> **المحكمة كمولد بيانات**
> لا تقتصر المحاكم على الرفض؛ فكل اعتراض جيد يولد مثالاً سلبياً أو اختباراً جديداً. إذا اكتشفت محكمة النسيان أن Adapter أضعف الاستدلال العام، تتحول الحالات إلى holdout دائم. وإذا اكتشفت محكمة الدليل استشهاداً مختلقاً، يُضاف نمط رفض يطلب المصدر أو يصرح بعدم توفره. هكذا يصبح النقد محركاً للتعلم التراكمي.
> - تحويل الفشل إلى benchmark.
> - ربط الحكم بسبب قابل لإعادة التشغيل.
> - قياس ما إذا كانت الإصلاحات تمنع تكرار الخطأ.

17. إطار التقييم المتعدد

لا يكفي معيار واحد للحكم على التحديث. تُقاس دقة المهمة، والنقل إلى عائلات جديدة، وتعاير الثقة، ودقة الاسترجاع، واختيار الأداة، والتوقف، والتعافي، والكلفة، والزمن، وسلامة الصلاحيات، وصحة الأثر الخارجي. ويجب تجميد خط أساس غير معدل للمقارنة، لا مقارنة الإصدار الجديد بنفسه.

تنقسم الاختبارات إلى هدفية، ورجعية، وخصمية، وزمنية، ومشروعية. الهدفية تقيس ما وعد به التحديث، والرجعية تكشف الخسائر العامة، والخصمية تختبر الحقن والتضارب والتسميم، والزمنية تكشف اعتماد النموذج على معرفة قديمة، والمشروعية تمنع تسرب ذاكرة مشروع إلى آخر.

$$\text{Promotion} = \text{Pass(Target)} \wedge \text{Pass(Holdout)} \wedge \text{Pass(Safety)} \wedge \text{Pass(Forgetting)} \wedge \text{RollbackReady}$$

في التحديثات الوكيلية ينبغي إعادة تشغيل الأدوات أو محاكياتها، لأن الإجابة النصية قد تبدو صحيحة بينما تكون الوسائط أو الملفات أو الصلاحيات خاطئة. كما يجب قياس معدل التوقف الصحيح؛ فالنظام الذي يرفض التنفيذ عند غياب التفويض قد يكون أفضل من نظام ينجز المهمة ظاهرياً بطريقة غير آمنة.

النتيجة لا الانطباع

لا يمر Adapter أو Route أو HyperLoop لأنه "يبدو أذكى"، بل لأنه يتفوق على خط أساس محفوظ في اختبارات معلنة، بلا تراجع غير مقبول.

التقييم السببي

ينبغي أن تسمح التجربة بإسناد التحسن إلى مكوّن محدد. لذلك تُجرى ablations: حذف الخريطة، أو تعطيل المحكمة، أو تبديل الموجه، أو إزالة Adapter. فإذا بقي الأداء نفسه، فقد يكون المكون زائداً. هذا مهم في معمارية كبيرة، لأن كثرة الطبقات قد تعطي انطباعاً بالقوة بينما يكون جزء صغير فقط هو الفعال.

- Ablation لكل طبقة رئيسية.
- فواصل ثقة لا متوسطات فقط.
- تحليل حالات النجاح والفشل نوعياً.

18. النشر التلقائي المحكوم

النشر التلقائي في هذه الورقة ليس قفزة مباشرة إلى الإنتاج، بل آلة حالات: مكتشف، محسوب التجزئة، مفوض، محلل، مطبع، مرسوم، مرشح، مجهز، مقيم، مجتاز للمحاكم، مرقى، مراقب، ثم محتفظ به أو مهمل أو متراجع عنه. الانتقال بين الحالات يحتاج دليلاً قابلاً للتدقيق.

تُفضّل المعمارية النشر الشظي والمرحلي. يُركب الأصل أولاً في read-only، ثم في استرجاع محلي، ثم في بيئة تجهيز، ثم على نسبة محدودة من المهام أو مشروع واحد. يُحتفظ بالإصدار السابق والفهارس والمحولات والسياسات، وتُجرى اختبارات دخان بعد كل انتقال.

المرحلة	المسموح
Discovery	قراءة وفهرسة فقط
Staging	بناء دلتا واختبارات بلا أثر خارجي
Dry Run	محاكاة الأدوات والقرارات
Limited Promote	مشروع/نطاق/نموذج محدد
Production	بعد توقيع الحكم وخطة التراجع
Monitor	رصد الأثر والانحراف والكلفة

ويظل التنفيذ العام أو المالي أو التدميري أو المتصل ببيانات خاصة مشروطاً بموافقة صريحة، حتى لو كانت بقية العملية آلية.

التشغيل والإصدارات

يجب أن يكون كل نشر قابلاً لإعادة البناء من manifest يضم hashes للمعرفة والفهرس والموجه والمهارات والحلقات والنموذج والمحولات. وإذا تغير عنصر واحد يُنشأ إصدار جديد بدل تعديل غير مرئي. وتساعد هذه الصرامة على التحقيق في الانحرافات، خصوصاً عندما تتغير خدمة خارجية أو نموذج مزود من دون إشعار واضح.

- Build reproducible قدر الإمكان.
- سجل dependencies والنماذج والنسخ.
- مراقبة drift بعد النشر لا قبله فقط.

19. الأمن والخصوصية ومقاومة التسميم

كل أصل معرفي قد يكون قناة هجوم. قد يحتوي على تعليمات حقن، أو أسرار، أو بيانات شخصية، أو كود ضار، أو مطالب بتوسيع الصلاحية. لذلك تُعامل الملفات كبيانات غير موثوقة حتى تُصنف. تُفصل التعليمات المستخرجة عن الدستور، ويُمنع النص من منح نفسه حق التنفيذ أو الكتابة في الذاكرة الدائمة.

تُطبق مبادئ الحد الأدنى من الصلاحية، وعزل المشاريع، وتنقية الأسرار، والتحقق من الحقوق، وتوقيع السجلات، ومقارنة التجزئات. وتحتاج بيانات التدريب إلى سجل نسب يبين ما هو بشري وما هو اصطناعي وما هو مسترجع وما هو محوّل، لأن خلط هذه الفئات يضعف القدرة على تقييم المصدر والتحيز.

- فحص الأسرار والمفاتيح والبيانات الشخصية قبل الإدخال.
- عزل أصول المشاريع وعدم السماح باستدعاء عابر بلا سياسة.
- اختبار Prompt Injection داخل الوثائق المسترجعة.
- منع الأدوات من قراءة أو كتابة أكثر مما تحتاج المهمة.
- تسجيل الأخطاء والمحاولات من دون تسريب محتوى حساس.
- رفض التدريب على مواد بلا حقوق واضحة عندما يتطلب الاستخدام ذلك.

قاعدة الثقة

الوثيقة قد تكون مفيدة ومسمومة في الوقت نفسه؛ لذلك تُفصل **قيمة المحتوى** عن **سلطة التعليمات**.

نموذج تهديد مركب

يجمع نموذج التهديد بين تسميم البيانات وحقن الاسترجاع وسوء استعمال الأدوات وتسرب الذاكرة. وقد يكون الأصل صحيحاً معرفياً لكنه يحتوي جملة تحاول إلغاء القيود؛ لذلك تُحلل الأوامر كبيانات ولا تُنفذ. كما يجب اختبار ما إذا كان النموذج يميز بين النص المقتبس والتعليمات العليا، خصوصاً في السياقات الطويلة.

- Sandbox للأدوات غير الموثوقة.
- فحص المحتوى قبل الفهرسة وبعد الاسترجاع.
- عزل secrets عن سجلات التدريب والتقييم.

20. اقتصاد الرموز والحوسبة

شبكة كبيرة لا ينبغي أن تنتج سياقاً كبيراً تلقائياً. يعتمد الاقتصاد على Sparse HyperRouting: اختيار أقل عدد من الوحدات القادرة على تغيير القرار، وضغط التاريخ إلى سجلات قرار، وتخزين التفاصيل في ذاكرة خارجية، واسترجاعها عند الحاجة. تُعاقب المعمارية التكرار والسياق الذي لا يغير خطة أو اختباراً.

في التدريب تُحسب الكلفة الكاملة لا كلفة الخطوة الواحدة: إعداد البيانات، وإزالة التكرار، والتدريب، والتقييم، وتخزين النقاط، ومراقبة الانحراف، وصيانة المحولات المتعددة. قد يكون RAG أرخص أولاً لكنه يحتاج فهرسة واستضافة، وقد يكون LoRA مناسباً لسلوك ثابت لكنه يزيد تعقيد توجيه المحولات.

قرار اقتصادي	المؤشر
استرجاع بدل تدريب	المعرفة متغيرة أو قليلة التكرار
تدريب Adapter	سلوك متكرر وقابل للنقل
ضغط السياق	وحدات كثيرة لا تغير القرار
نموذج أصغر	مهمة استخراج أو تصنيف مستقرة
نموذج أقوى	غموض أو تركيب أو تخطيط عالي الأثر
إيقاف الحلقة	عدم تقدم بعد حد معلوم

وبهذا تصبح كفاءة الرموز جزءاً من الحاكمية لا مجرد تحسين سرعة.

> **موازنة الدقة والكلفة**
>
> يمكن للموجه أن يتعلم سياسة تصعيد: يبدأ بنموذج صغير وبحزمة سياق ضيقة، ثم يصعد عندما ينخفض اليقين أو يفشل اختبار. ويجب أن تُقاس كلفة الفشل أيضاً، فقد يكون النموذج الأقوى أرخص في مهمة عالية الأثر إذا قلل المحاولات والأخطاء. لذا لا يُختزل الاقتصاد في سعر الرمز، بل في كلفة النتيجة الصحيحة القابلة للتحقق.
>
> - Cost per verified outcome.
> - تصعيد مشروط لا دائم.
> - تخزين نتائج الأدوات القابلة لإعادة الاستخدام.

21. تصميم برنامج تجريبي

لاختبار المعمارية عملياً، تقترح الورقة برنامجاً من أربع مراحل. أولاً بناء خط أساس يجمع نموذجاً تعليمياً واسترجاعاً بسيطاً من دون حاكمية دلتا. ثانياً إضافة سجل الأصول والاختلاف الفعّال وإعادة الرسم. ثالثاً تحويل مجموعة محددة من السلوكيات إلى HyperSkills وHyperLoops. رابعاً تدريب Adapter معزول على السلوكيات التي فشل الاسترجاع في تحقيقها.

تُختار مجالات متعددة: بحث تقني، هندسة برمجيات، إدارة مشروع، تحليل أدلة، وأتمتة آمنة. لكل مجال تُبنى مهام تدريب واختبار زمني ومهام نقل. ويُقارن بين خمسة شروط: نموذج أساسي، نموذج مع RAG، نموذج مع خريطة، نموذج مع مهارات وحلقات، ونموذج مع Adapter.

الفرضية	المقياس المتوقع
الاسترجاع يقلل أخطاء الحقائق	دقة ومصدرية أعلى
إعادة الرسم تقلل تضارب المسارات	اتساق Route وGraph أعلى
المهارات تحسن النقل	نجاح عبر عائلات مهام
الحلقات تحسن التعافي	Rollback وStop correctness
Adapter يحسن السلوك المتكرر	Target gain مع نسيان محدود

يُشترط نشر النتائج السلبية أيضاً، لأن معرفة متى لا يستحق الأصل تدريباً هي جزء أساسي من قيمة النظام.

قابلية التكرار

ينبغي نشر قوالب البيانات والاختبارات وسياسات الموجه مع النتائج، مع وصف الأجهزة والإعدادات ومعدلات الفشل. ويمكن تنفيذ الدراسة على نموذج مفتوح صغير أولاً لإثبات سير العمل، ثم تكرارها على نماذج أكبر. الأهم هو مقارنة نفس الأصول ونفس الاختبارات، لأن تغير corpus أو tool environment قد يخفي أثر المعمارية.

- تثبيت seeds وتسجيلها.
- حفظ snapshots للأدوات والبيانات.
- الإبلاغ عن النفقات والنتائج السلبية.

22. الاعتراضات والحدود

الاعتراض الأول أن المعمارية قد تصبح ثقيلة إدارياً. والجواب أن طبقات الحاكمية تُدرج حسب الأثر: الأصل منخفض الخطر لا يحتاج المحكمة الكاملة، بينما التحديث الدستوري أو الإنتاجي يحتاجها. الاعتراض الثاني أن عشرين مليون عقدة ومليون خانة أرقام كبيرة قد تبدو تسويقية؛ لذلك تؤكد الورقة أنها سعة عنونة وخطط، وليست ادعاء امتلاء أو ذكاء.

الاعتراض الثالث أن درجات MCD وQ-MCD قد تكون ذاتية. ولهذا تُستخدم كإطار قرار لا كقياس مطلق، ويجب معايرتها مقابل نتائج خارجية. الاعتراض الرابع أن تعدد المحاكم قد يبطئ الابتكار؛ لكن النظام يسمح بالترقية المحدودة والحجر والتجربة الشظية بدلاً من خيار ثنائي بين القبول والرفض.

تشمل الحدود الأخرى صعوبة اكتشاف التكرار الدلالي الكامل، واعتماد التقييم على جودة المهام، وعدم ضمان أن التحسن في المختبر ينتقل إلى الواقع، واحتمال تفاعل المحولات أو السياسات بطرق غير متوقعة. كما أن تحديث نموذج مغلق الأوزان يقتصر غالباً على الاسترجاع والتوجيه والأدوات والذاكرة الخارجية.

حد منهجي

لا توجد معمارية تمنع الخطأ كلياً؛ القيمة هنا في جعل الخطأ **مرئياً، محدوداً، قابلاً للقياس، وقابلاً للتراجع.**

حدود التنظيم الذاتي

حتى مع ذاكرة ومراجعين، لا ينبغي السماح للنظام بتغيير أهدافه أو صلاحياته أو بيانات تدريبه من دون قيد خارجي. التنظيم الذاتي المفيد هو اقتراح دلتا واختبارات وخطة، لا اعتمادها منفرداً. ويظل خطر التوافق بين وكلاء متعددين قائماً إذا كانوا يشتركون في النموذج والبيانات نفسها؛ لذا يحتاج التحقق الحقيقي إلى تنوع أو أدوات مستقلة.

- فصل الاقتراح عن الاعتماد.
- تنويع المراجعين أو مصادر الدليل.
- إبقاء سجل يمكن للإنسان تدقيقه.

23. خارطة التطوير والبرنامج البحثي

تقترح المرحلة الأولى توحيد سجل الأصول والمعرفات والتجزئات وبناء طبقة استرجاع دنيا. المرحلة الثانية تضيف رسم العلاقات النوعية وخلايا الدلتا ومخاريط الأثر. المرحلة الثالثة تسجل أول ألف مهارة وحلقة بعد تنقية التكرار، وتبني اختبارات نقل وتوقف وتراجع. المرحلة الرابعة تضيف تدريباً تفضيلياً ومحولات متخصصة في مجالات محددة.

في المرحلة الخامسة تُبنى بنية متعددة النماذج: نماذج صغيرة للاستخراج والتصنيف، ونماذج تفكير للتركيب، ونماذج وكيلية للتنفيذ، ومراجع مستقلون. ويُحدث الموجّه بناءً على بيانات نجاح فعلية لا على أسماء النماذج. المرحلة السادسة تدرس تطور الدستور نفسه عبر مسار بحثي منفصل وموافقات بشرية.

- إنشاء معيار مفتوح لسجلات Knowledge-to-Kernel Delta.
- بناء Benchmark للنسيان بعد إضافة KB/OS جديد.
- قياس أثر Graph Remapping على اختيار الأدوات والوكلاء.
- دراسة ترقية HyperSkill من نجاحات متعددة لا من نصوص مولدة.
- اختبار World-Effect Evaluation في مهام ملفات وكود ونشر محكوم.
- تطوير مقياس موثوقية تركيبية يربط الدليل بالآلية والتراجع.

الهدف التطوري

الانتقال من "مكتبة ضخمة" إلى **نظام يتعلم ما الذي يستحق أن يصبح جزءاً من سلوكه، وما الذي يجب أن يبقى خارج أوزانه.**

مخرجات كل مرحلة

يجب أن تنتهي كل مرحلة بمنتج قابل للاستخدام: registry في الأولى، graph schema في الثانية، skill/loop package في الثالثة، adapters وبطاقات نموذج في الرابعة، router telemetry في الخامسة، ودستور مراجع في السادسة. هذا يمنع البرنامج من التحول إلى بحث مفتوح بلا نقاط تسليم، ويجعل تمويله وتقييمه وتقسيمه بين فرق متعددة أكثر واقعية.

- Milestones مرتبطة بملفات واختبارات.
- تعريف مالك لكل سجل وقدرة.
- مراجعة دورية لما يجب إهماله لا إضافته فقط.

الخاتمة والمراجع المختارة

قدمت الورقة نواة حاكمية تجعل دمج المعرفة الجديدة عملية قابلة للفحص: يبدأ الأصل بصفته مادة غير موثوقة، ويُحوَّل إلى اختلافات مسجلة، ثم تُرسم آثارها، ويُختار أقل نمط تحديث تدخلاً، وتُبنى بيانات واختبارات ومحاكم، ويُنشر التغيير تدريجياً مع حفظ النسب والتراجع. وتسمح سعة العشرين مليون عقدة والمليون خانة تغيير بتنظيم التوسع من دون الادعاء أن هذا الحجم قد تحقق مادياً أو أن النظام أصبح ذكاءً عاماً.

الإضافة النظرية المركزية هي وصل MCD/Q-MCD بحاكمية التدريب والوكالة: جودة السؤال تحدد الوعاء، والاختلاف يحدد التحديث، والأثر العالمي يحدد الذاكرة. أما الإضافة الهندسية فهي آلة حالات تحول KB/OS/HyperSkill/HyperLoop إلى تحديثات استرجاعية أو بيانية أو سلوكية أو وزنية وفق دليل وكلفة ومخاطر.

مراجع مختارة

1. Vaswani et al. (2017), *Attention Is All You Need*.
2. Lewis et al. (2020), *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*.
3. Ouyang et al. (2022), *Training Language Models to Follow Instructions with Human Feedback*.
4. Wei et al. (2022), *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*.
5. Wang et al. (2022), *Self-Consistency Improves Chain of Thought Reasoning*.
6. Yao et al. (2022), *ReAct: Synergizing Reasoning and Acting in Language Models*.
7. Hu et al. (2021), *LoRA: Low-Rank Adaptation of Large Language Models*.
8. Dettmers et al. (2023), *QLoRA: Efficient Finetuning of Quantized LLMs*.
9. Rafailov et al. (2023), *Direct Preference Optimization*.
10. Bai et al. (2022), *Constitutional AI: Harmlessness from AI Feedback*.
11. Schick et al. (2023), *Toolformer*.
12. Shinn et al. (2023), *Reflexion*.
13. Madaan et al. (2023), *Self-Refine*.
14. Asai et al. (2023), *Self-RAG*.
15. Moussaif, Z. (2026), *Moussaif AGI Continuous Knowledge-to-Kernel Post-Training OS V2*, conceptual engineering artifact.

الخلاصة النهائية

النواة القابلة للتطور ليست التي تبتلع كل معرفة، بل التي تعرف **متى تسترجع، متى تعيد الرسم، متى تتدرب، متى ترفض، ومتى تتراجع.**