mcai-v2 agi new technology neoAGI

هندسة حاكمية وعائية لتجاوز نماذج التفكير الحالية نحو ذكاء عام اصطناعي قابل للاختبار

زهير مساعف / Zouhair Moussaif

25 يونيو 2026

نسخة بحثية عربية RTL بخط Amiri - حاكمية، تفنيد، ذاكرة، عالم سببي، ووكالة قابلة للاختبار

فهرس تفصيلي

- mcai-v2 agi new technology neoAGI
  - هندسة حاكمية وعائية لتجاوز نماذج التفكير الحالية نحو ذكاء عام اصطناعي قابل للاختبار
  - تنبيه منهجي حول مراتب الدليل
  - ملخص تنفيذي من صفحتين
    - خريطة مفاهيم نصية
  - سجل مصادر داخلي
  - جدول الثابت/الراجح/الخلافي/الفرضيات
  - الباب الأول: تشخيص حدود بنية النماذج المفكرة الحالية
    - 1.1 لماذا test-time compute ليس AGI؟
    - 1.2 مشكلة الذكاء المتعرج jagged intelligence
    - 1.3 التفكير النصي دون حاكمية واقعية
    - 1.4 من RAG إلى GraphRAG: أين يبقى العمى؟
    - 1.5 الذاكرة في الوكلاء: بين المخزن والتاريخ الداخلي
    - 1.6 الخطأ المقنّع في benchmarks
    - 1.7 حدود التخطيط عند LLM agents
    - 1.8 الفرق بين مهارة محلولة وذكاء مكتسب
  - الباب الثاني: التأسيس المفاهيمي لـ MCAI-v2
    - 2.1 الذكاء بوصفه حاكمية للفروق لا تجميع إجابات
    - 2.2 MCD كمقياس فرز للتعقيد المنتج
    - 2.3 نظرية التراكب الوعائي: استعمالها كعدسة لا كدليل نهائي
    - 2.4 الحاكمية الإبستمولوجية قبل الجواب
    - 2.5 الانتقائية المساعفية كإنقاذ مشروط للمسار
    - 2.6 الزمن الداخلي للمعرفة
    - 2.7 الاختطاف الوعائي والانتقال بين الأوعية
    - 2.8 من النموذج اللغوي إلى محرك معرفة حاكم
  - الباب الثالث: النظريات الثلاث الجديدة
    - 3.1 نظرية التعميم الوعائي المتمايز VDG-Moussaif
    - 3.2 نظرية الزمن الداخلي السببي ITC-Moussaif
    - 3.3 نظرية المناعة الإبستمولوجية EIG-Moussaif
    - 3.4 كيف تتكامل النظريات الثلاث؟
    - 3.5 ما الذي تضيفه فوق CoALA وReAct وToT؟
    - 3.6 الشروط التي تجعل النظرية قابلة للهدم
    - 3.7 الحد الأدنى الرياضي للنظريات
    - 3.8 حدود الادعاء: فرضية تشغيلية لا ضمان AGI
  - الباب الرابع: التقنيات الثلاث الجديدة
    - 4.1 تقنية HAKIM-MCD v2 Operating Layer
    - 4.2 تقنية Vascular GraphWorld Memory
    - 4.3 تقنية Falsification-Centric Test-Time Compute
    - 4.4 هندسة الربط بين الذاكرة والعالم والقرار
    - 4.5 تنزيل تقني على وكيل برمجي
    - 4.6 تنزيل تقني على باحث معرفي
    - 4.7 تنزيل تقني على روبوت/بيئة تفاعلية
    - 4.8 جدول مخاطر وتخفيف
  - الباب الخامس: الأدوات الخمس عشرة لمحرّك neoAGI
    - 5.1 Critical Difference Extractor
    - 5.2 MCD Meta-Complexity Mapper
    - 5.3 Vascular Compute Router
    - 5.4 Inner-Time Budgeter
    - 5.5 Causal Counterfactual Forge
    - 5.6 GraphWorld Builder
    - 5.7 Epistemic Immune Filter
    - 5.8 Perturbation Court
    - 5.9 Moussaif Selective Rescue
    - 5.10 Novelty Entropy Meter
    - 5.11 Anti-Hijack Source Ledger
    - 5.12 Skill-to-Hyperskill Compiler
    - 5.13 World-Effect Loop Monitor
    - 5.14 Self-Reconstruction Memory
    - 5.15 Benchmark Tribunal
  - الباب السادس: فلسفات التدريب وأدوات التدريب
    - 6.1 تدريب الفروق لا الإجابات
    - 6.2 تدريب التفنيد قبل الثقة
    - 6.3 تدريب الذاكرة كإعادة بناء لا كاسترجاع
    - 6.4 تدريب العالم السببي لا النص وحده
    - 6.5 تدريب الانتقاء المساعفي
    - 6.6 أداة Vascular Curriculum Generator
    - 6.7 أداة Failure Distillation Lab
    - 6.8 أداة Causal Replay Forge
    - 6.9 أداة Benchmark Anti-Gaming Lab
    - 6.10 أداة Governance Reward Model
  - الباب السابع: البنية التقنية الكاملة
    - 7.1 مخطط طبقات MCAI-v2
    - 7.2 واجهات النظام APIs
    - 7.3 صيغة الذاكرة والتجربة
    - 7.4 بروتوكول الاسترجاع الحاكم
    - 7.5 بروتوكول التخطيط الوعائي
    - 7.6 بروتوكول التفنيد المتعدد
    - 7.7 بروتوكول التعلّم من الأثر
    - 7.8 النشر على Agentic OS
  - الباب الثامن: القياس والاختبار والـbenchmarks
    - 8.1 لماذا لا نلفّق درجات benchmark؟
    - 8.2 مصفوفة 13 معيارا للاختبار
    - 8.3 ARC-AGI كاختبار كفاءة اكتساب مهارة
    - 8.4 SWE-bench كاختبار أثر واقعي
    - 8.5 GAIA/WebArena/OSWorld/Terminal-Bench كاختبارات وكيل
    - 8.6 مقياس VGB الحاكمي
    - 8.7 اختبار مقاومة تلوث البيانات
    - 8.8 اختبار قابلية الهدم والتعميم

- الباب التاسع: النموذج الجديد M-VCGE
  - 9.1 تعريف Moussaif Vascular-Causal Governance Engine
  - 9.2 طبقات النموذج
  - 9.3 آليات القرار
  - 9.4 ما يفسره أفضل من النماذج الحالية
  - 9.5 ما لا يفسره
  - 9.6 تنبؤات قابلة للاختبار
  - 9.7 نسخة صفحة واحدة
  - 9.8 خريطة تشغيل
- الباب العاشر: محكمة الهدم والنسخة الأقوى
  - 10.1 أقوى 10 اعتراضات
  - 10.2 أقوى 10 نقاط ضعف
  - 10.3 ما قد يكون مبالغا فيه
  - 10.4 ما يحتاج إلى دليل أكثر
  - 10.5 ماذا تقول المدرسة المنافسة؟
  - 10.6 ما الذي يفند MCAI-v2؟
  - 10.7 النسخة المعدلة الأقوى
  - 10.8 سجل البناء المعرفي
- الباب الحادي عشر: البرنامج البحثي المستقبلي
  - 11.1 12 فرضية بحثية
  - 11.2 12 سؤالا مستقبليا
  - 11.3 12 مشروعا قابلا للتنفيذ
  - 11.4 12 مفهوما يجب ضبطه
  - 11.5 12 خطأ شائعا
  - 11.6 12 تمييزا معرفيا
  - 11.7 خارطة 24 شهرا
  - 11.8 خاتمة تقنية
- ملحق الأدوات التنفيذية الخمس عشرة - مواصفات قابلة للبرمجة
- فلسفات التدريب الخمس
  - تدريب الفروق لا الإجابات
  - تدريب التفنيد قبل الثقة
  - تدريب الذاكرة كإعادة بناء
  - تدريب العالم السببي
  - تدريب الانتقاء المساعفي
- أدوات التدريب الخمس
  - Vascular Curriculum Generator
  - Failure Distillation Lab
  - Causal Replay Forge
  - Benchmark Anti-Gaming Lab
  - Governance Reward Model
- مصفوفة 13 benchmark دون تلفيق درجات
- النموذج الجديد: M-VCGE
  - محكمة الهدم: أقوى الاعتراضات والجواب المعدل
  - قاموس مصطلحات
- 12 فرضية بحثية قوية
- 12 سؤالا مستقبليا
- 12 مشروعا بحثيا قابلا للتنفيذ
- ملحق بطاقات البروتوكولات التقنية المكثفة
  - بطاقة 1: بروتوكول منع الهلوسة
  - بطاقة 2: بروتوكول اختيار الأداة
  - بطاقة 3: بروتوكول ترقية الذاكرة
  - بطاقة 4: بروتوكول ضغط التجربة
  - بطاقة 5: بروتوكول قياس الجدة
  - بطاقة 6: بروتوكول محاكمة المصدر
  - بطاقة 7: بروتوكول التخطيط السببي
  - بطاقة 8: بروتوكول تحويل المهارة إلى hyperskill
  - بطاقة 9: بروتوكول مقاومة benchmark gaming
  - بطاقة 10: بروتوكول بناء GraphWorld
  - بطاقة 11: بروتوكول التفنيد العكسي
  - بطاقة 12: بروتوكول أثر العالم
  - بطاقة 13: بروتوكول اختيار الفشل المفيد
  - بطاقة 14: بروتوكول الخريطة الوعائية
  - بطاقة 15: بروتوكول تحديث reward governance
  - بطاقة 16: بروتوكول اختبار few-shot transfer
  - بطاقة 17: بروتوكول قياس تكلفة الخطأ
  - بطاقة 18: بروتوكول استعمال مصادر متعارضة
  - بطاقة 19: بروتوكول المعرفة الزمنية
  - بطاقة 20: بروتوكول الإصدار والتجريب
  - بطاقة 21: بروتوكول منع الهلوسة
  - بطاقة 22: بروتوكول اختيار الأداة
  - بطاقة 23: بروتوكول ترقية الذاكرة
  - بطاقة 24: بروتوكول ضغط التجربة
  - بطاقة 25: بروتوكول قياس الجدة
  - بطاقة 26: بروتوكول محاكمة المصدر
  - بطاقة 27: بروتوكول التخطيط السببي
  - بطاقة 28: بروتوكول تحويل المهارة إلى hyperskill
  - بطاقة 29: بروتوكول مقاومة benchmark gaming
  - بطاقة 30: بروتوكول بناء GraphWorld
  - بطاقة 31: بروتوكول التفنيد العكسي
  - بطاقة 32: بروتوكول أثر العالم
  - بطاقة 33: بروتوكول اختيار الفشل المفيد
  - بطاقة 34: بروتوكول الخريطة الوعائية
  - بطاقة 35: بروتوكول تحديث reward governance
  - بطاقة 36: بروتوكول اختبار few-shot transfer
  - بطاقة 37: بروتوكول قياس تكلفة الخطأ
  - بطاقة 38: بروتوكول استعمال مصادر متعارضة
  - بطاقة 39: بروتوكول المعرفة الزمنية
  - بطاقة 40: بروتوكول الإصدار والتجريب
  - بطاقة 41: بروتوكول منع الهلوسة
  - بطاقة 42: بروتوكول اختيار الأداة
  - بطاقة 43: بروتوكول ترقية الذاكرة
  - بطاقة 44: بروتوكول ضغط التجربة
  - بطاقة 45: بروتوكول قياس الجدة
  - بطاقة 46: بروتوكول محاكمة المصدر
  - بطاقة 47: بروتوكول التخطيط السببي
  - بطاقة 48: بروتوكول تحويل المهارة إلى hyperskill
  - بطاقة 49: بروتوكول مقاومة benchmark gaming

- بطاقة 50: بروتوكول بناء GraphWorld
- بطاقة 51: بروتوكول التفنيد العكسي
- بطاقة 52: بروتوكول أثر العالم
- بطاقة 53: بروتوكول اختيار الفشل المفيد
- بطاقة 54: بروتوكول الخريطة الوعائية
- بطاقة 55: بروتوكول تحديث reward governance
- بطاقة 56: بروتوكول اختبار few-shot transfer
- بطاقة 57: بروتوكول قياس تكلفة الخطأ
- بطاقة 58: بروتوكول استعمال مصادر متعارضة
- بطاقة 59: بروتوكول المعرفة الزمنية
- بطاقة 60: بروتوكول الإصدار والتجريب
- بطاقة 61: بروتوكول منع الهلوسة
- بطاقة 62: بروتوكول اختيار الأداة
- بطاقة 63: بروتوكول ترقية الذاكرة
- بطاقة 64: بروتوكول ضغط التجربة
- بطاقة 65: بروتوكول قياس الجدة
- بطاقة 66: بروتوكول محاكمة المصدر
- بطاقة 67: بروتوكول التخطيط السببي
- بطاقة 68: بروتوكول تحويل المهارة إلى hyperskill
- بطاقة 69: بروتوكول مقاومة benchmark gaming
- بطاقة 70: بروتوكول بناء GraphWorld
- بطاقة 71: بروتوكول التفنيد العكسي
- بطاقة 72: بروتوكول أثر العالم
- بطاقة 73: بروتوكول اختيار الفشل المفيد
- بطاقة 74: بروتوكول الخريطة الوعائية
- بطاقة 75: بروتوكول تحديث reward governance
- بطاقة 76: بروتوكول اختبار few-shot transfer
- بطاقة 77: بروتوكول قياس تكلفة الخطأ
- بطاقة 78: بروتوكول استعمال مصادر متعارضة
- بطاقة 79: بروتوكول المعرفة الزمنية
- بطاقة 80: بروتوكول الإصدار والتجريب
- بطاقة 81: بروتوكول منع الهلوسة
- بطاقة 82: بروتوكول اختيار الأداة
- بطاقة 83: بروتوكول ترقية الذاكرة
- بطاقة 84: بروتوكول ضغط التجربة
- بطاقة 85: بروتوكول قياس الجدة
- بطاقة 86: بروتوكول محاكمة المصدر
- بطاقة 87: بروتوكول التخطيط السببي
- بطاقة 88: بروتوكول تحويل المهارة إلى hyperskill
- تقييم الجودة النهائي

تنبيه منهجي حول مراتب الدليل

لا يدعي هذا الكتاب أن تقنية واحدة تضمن الوصول الحتمي إلى AGI. العبارة الهجومية المطلوبة تفهم هنا تخطة قصوى: ما التقنيات التي لو ثبتت تجريبيا ستجعل الطريق إلى AGI أقرب من هندسات النماذج المفكرة الحالية؟ لذلك يفصل الكتاب بين: الثابت، الراجح، الخلافي، الفرضية، والتركيب الفلسفي.

الرتبة	المعنى	طريقة التعامل
ثابت/مدعوم	نتيجة أو اتجاه له سند بحثي قوي	يستعمل كأساس لا كبرهان AGI
راجح	اتجاه تؤيده أدلة متعددة مع حدود	يعتمد مع احتراز
خلافي	تفسير متنازع	يعرض مع اعتراضاته
فرضية MCAI-v2	اختراع أو تركيب قابل للاختبار	لا يطرح كحقيقة حتى ينجح prototype
تأمل فلسفي	عدسة توليدية	يسمح بها فقط إذا أنتجت آلية أو اختبارا

ملخص تنفيذي من صفحتين

أطروحة الكتاب: البنية التي تقربنا من AGI لن تكون نموذجا لغويا أكبر فقط، ولا سلسلة agentic loops أكثر ضجيجا، بل نظام حاكمية معرفية يوزع التفكير والاسترجاع والأدوات والذاكرة والتفنيد وفق قيمة الفرق المنتج. أحدث النماذج المفكرة بينت أن RL وtest-time compute يرفعان القدرة في مجالات محددة، وأن الوكلاء مع tools/memory يستطيعون إنجاز مسارات أطول، لكن الفجوة الحاسمة تبقى في التعميم القليل الخبرة، سلامة الحكم، مقاومة التلاعب بالاختبارات، وبناء ذاكرة أثر تتعلم من العالم لا من النص وحده.

يقترح الكتاب MCAI-v2 كطبقة فوق-نموذجية: HAKIM-MCD v2 يحكم متى يفكر النظام، Vascular GraphWorld Memory يحول المعرفة إلى عالم علائقي سببي، وFalsification-Centric Test-Time Compute يجعل وقت الاستدلال محكمة احتمالات لا سلسلة جمل.

النظرية ليست إلغاء للLLM؛ بل إعادة تحديد دوره: LLM مولد ومفسر ومحول لغوي، بينما الحاكمية تقرر أي مسار يستحق الإنفاق، أي مصدر يستحق الثقة، أي ذاكرة تستحق الترقية، وأي فشل يستحق الإنقاذ.

الإضافة الاختراعية في الكتاب تتكون من ثلاث نظريات: التعميم الوعائي المتمايز، الزمن الداخلي السببي، والمناعة الإبستمولوجية؛ وثلاث تقنيات: Vascular GraphWorld Memory، HAKIM-MCD v2 Operating Layer، وFalsification-Centric Test-Time Compute؛ وخمس عشرة أداة تنفيذية تبدأ من Critical Difference Extractor وتنتهي بـBenchmark Tribunal. كل أداة مصممة كي تكون قابلة للبرمجة والاختبار، لا مجرد مصطلح.

معيار النجاح ليس أن نعلن AGI، بل أن نثبت نمطا: تحسن عبر جلسات متعاقبة، نقل مهارة إلى مجال جديد، انخفاض الهلوسة، مقاومة التلوث، وارتفاع القدرة على حل مهام قليلة الأمثلة مثل ARC-AGI أو مهام فعلية مثل SWE-bench Verified دون استغلال benchmark. لذلك يرفض الكتاب اختلاق درجات benchmarks، ويقترح بدلا من ذلك بروتوكول تقييم من 13 معيارا، وVGB-Moussaif لاختبار الحاكمية نفسها.

خريطة مفاهيم نصية

```
[مصادر] -> [Source Ledger] -> [GraphWorld] -> [MCD Mapper] -> [Vascular Router]
-> {استرجاع | تفكير | أداة | محاكاة | تفنيد}
-> [Perturbation Court] -> [Selective Rescue] -> [World-Effect Loop]
-> [Self-Reconstruction Memory] -> [Hyperskill Compiler] -> [Benchmark
Tribunal]
```

سجل مصادر داخلي

رمز	المصدر	المؤلف/المؤسسة	السنة	أثره في الكتاب
R1	OpenAI o1 System Card	OpenAI	2024	نموذج مفكر، RL، تقييمات سلامة، حدود reasoning systems
R2	DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning	DeepSeek-AI	2025	RL reasoning، test-time reasoning، distillation، حدود readability/language mixing
R3	ReAct: Synergizing Reasoning and Acting in Language Models	Yao et al.	2022/2023	التناوب بين reasoning/actions واستعمال البيئة لتقليل الهلوسة
R4	Tree of Thoughts: Deliberate Problem Solving with Large Language Models	Yao et al.	2023	بحث شجري على وحدات فكرية، self-evaluation، backtracking
R5	Reflexion: Language Agents with Verbal Reinforcement Learning	Shinn et al.	2023	تعلم لغوي من الفشل عبر ذاكرة انعكاسية دون تحديث الأوزان
R6	From Local to Global: A Graph RAG Approach to Query-Focused Summarization	Edge et al. / Microsoft Research	2024	استخراج knowledge graph وcommunity summaries لتحسين RAG على corpora خاصة
R7	Cognitive Architectures for Language Agents - CoALA	Sumers et al.	2023/2024	ذاكرة معيارية، decision process، action space، هندسة وكلاء لغويين
R8	A Survey on the Memory Mechanism of Large Language Model based Agents	Zhang et al.	2024	أنماط ذاكرة الوكلاء، التقييم، التطبيقات، حدود الذاكرة الحالية
R9	Understanding the planning of LLM agents: A survey	Huang et al.	2024	تخطيط الوكلاء: selection، task decomposition، external modules، reflection، memory
R10	Large Language Models for Planning: A Comprehensive and Systematic Survey	Cao et al.	2025	تصنيف تخطيط LLM بين finetuning، modules، search-based methods
R11	Understanding World or Predicting Future? A Comprehensive Survey of World Models	Ding et al.	2024/2025	world models للفهم والتنبؤ، multimodal foundation models، evaluation fragmentation
R12	Towards Causal Representation Learning	Schölkopf et al.	2021	الربط بين السببية والتعميم والتحويل، واكتشاف المتغيرات السببية من observations
R13	On the Measure of Intelligence	François Chollet	2019	تعريف الذكاء ككفاءة اكتساب مهارة تحت priors وخبرة، ARC
R14	ARC-AGI-2 and ARC Prize 2025 Technical Report	ARC Prize / Chollet et al.	2025/2026	قياس few-shot generalization، تقرير 2025: أعلى نتيجة خاصة 24% على ARC-AGI-2
R15	SWE-bench: Can Language Models Resolve Real-World GitHub Issues?	Jimenez et al.	2023/2024	اختبار هندسة برمجيات واقعي من GitHub؛ أساس SWE-bench Verified
R16	Evaluating Frontier Models for Dangerous Capabilities	Phuong et al.	2024	منهج تقييم القدرات الخطرة: persuasion/ deception/cyber/self-proliferation/self-reasoning
R17	OpenAI - Introducing SWE-bench Verified	OpenAI	2024	نسخة بشرية التحقق من SWE-bench بـ 500 مسألة أكثر موثوقية
R18	GraphRAG Documentation	Microsoft	2024-2026	Pipeline: graph extraction, community hierarchy, summaries, retrieval over graph structures

جدول الثابت/الراجح/الخلافي/الفرضيات

الرتبة	الدعوى	الحد المعرفي	السند
ثابت/مدعوم بقوة	النماذج اللغوية يمكن تحسين أدائها بإضافة بحث وقت الاستدلال، ذاكرة، أدوات، واسترجاع منظم.	مدعوم بأعمال ReAct/ToT/Reflexion/ GraphRAG/CoALA.	R3-R8
راجح علميا	test-time compute وRL reasoning يرفعان أداء الرياضيات والبرمجة، لكنهما لا يثبتان AGI.	o1 وDeepSeek-R1 يبرزان الاتجاه وحدوده.	R1/R2
خلافي	هل chain-of-thought يمثل تفكيرا حقيقيا أم آثارا نصية مفيدة؟	الخلاف يبقى لأن السلوك ناجح محليا وفاشل/ متعرج عالميا.	R1/R2
فرضية الكتاب	AGI أقرب إلى حاكمية فروق وذاكرة عالمية سببية ذات تفنيد ذاتي، لا إلى نموذج أكبر فقط.	يحتاج prototype وbenchmarks ديناميكية.	MCAI-v2
تركيب فلسفي	الوعي/الذكاء كتركيب أوعية للمعنى والغاية والأثر.	عدسة توليدية من مشروع التراكب الوعائي، لا حقيقة مثبتة.	Moussaif

الباب الأول: تشخيص حدود بنية النماذج المفكرة الحالية

هذا الباب يبني طبقة 1 من معمارية neoAGI agi new technology mcai-v2. لا يشرح المفاهيم بوصفها درسا، بل يحولها إلى قرارات تصميم: ما الذي ندخله في الذاكرة؟ ما الذي نهدمه؟ ما الذي نقيسه؟ وما الذي يمنع الادعاء من التحول إلى دعاية؟

1.1 لماذا test-time compute ليس AGI؟

لماذا test-time compute ليس AGI؟ لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R1] [R2]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

1.2 مشكلة الذكاء المتعرج jagged intelligence

مشكلة الذكاء المتعرج jagged intelligence لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

1.3 التفكير النصي دون حاكمية واقعية

التفكير النصي دون حاكمية واقعية لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا، أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

1.4 من RAG إلى GraphRAG: أين يبقى العمى؟

من RAG إلى GraphRAG: أين يبقى العمى؟ لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R6] [R18]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا، أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

1.5 الذاكرة في الوكلاء: بين المخزن والتاريخ الداخلي

الذاكرة في الوكلاء: بين المخزن والتاريخ الداخلي لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R7] [R8]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

1.6 الخطأ المقنّع في benchmarks

الخطأ المقنّع في benchmarks لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

1.7 حدود التخطيط عند LLM agents

حدود التخطيط عند LLM agents لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R9] [R10]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

1.8 الفرق بين مهارة محلولة وذكاء مكتسب

الفرق بين مهارة محلولة وذكاء مكتسب لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R13] [R14]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

الباب الثاني: التأسيس المفاهيمي لـ MCAI-v2

هذا الباب يبني طبقة 2 من معمارية mcai-v2 agi new technology neoAGI. لا يشرح المفاهيم بوصفها درسا، بل يحولها إلى قرارات تصميم: ما الذي ندخله في الذاكرة؟ ما الذي نهدمه؟ ما الذي نقيسه؟ وما الذي يمنع الادعاء من التحول إلى دعاية؟

2.1 الذكاء بوصفه حاكمية للفروق لا تجميع إجابات

الذكاء بوصفه حاكمية للفروق لا تجميع إجابات لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

2.2 MCD كمقياس فرز للتعقيد المنتج

MCD كمقياس فرز للتعقيد المنتج لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

2.3 نظرية التراكب الوعائي: استعمالها كعدسة لا كدليل نهائي

نظرية التراكب الوعائي: استعمالها كعدسة لا كدليل نهائي لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R7] [R4] [R3]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا، أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

2.4 الحاكمية الإبستمولوجية قبل الجواب

الحاكمية الإبستمولوجية قبل الجواب لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا، أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

2.5 الانتقائية المساعفية كإنقاذ مشروط للمسار

الانتقائية المساعفية كإنقاذ مشروط للمسار لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

2.6 الزمن الداخلي للمعرفة

الزمن الداخلي للمعرفة لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

2.7 الاختطاف الوعائي والانتقال بين الأوعية

الاختطاف الوعائي والانتقال بين الأوعية لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

2.8 من النموذج اللغوي إلى محرك معرفة حاكم

من النموذج اللغوي إلى محرك معرفة حاكم لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

الباب الثالث: النظريات الثلاث الجديدة

هذا الباب يبني طبقة 3 من معمارية neoAGI mcai-v2 agi new technology. لا يشرح المفاهيم بوصفها درسا، بل يحولها إلى قرارات تصميم: ما الذي ندخله في الذاكرة؟ ما الذي نهدمه؟ ما الذي نقيسه؟ وما الذي يمنع الادعاء من التحول إلى دعاية؟

3.1 نظرية التعميم الوعائي المتمايز VDG-Moussaif

نظرية التعميم الوعائي المتمايز VDG-Moussaif لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

3.2 نظرية الزمن الداخلي السببي ITC-Moussaif

نظرية الزمن الداخلي السببي ITC-Moussaif لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

3.3 نظرية المناعة الإبستمولوجية EIG-Moussaif

نظرية المناعة الإبستمولوجية EIG-Moussaif لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

3.4 كيف تتكامل النظريات الثلاث؟

كيف تتكامل النظريات الثلاث؟ لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

3.5 ما الذي تضيفه فوق CoALA وReAct وToT؟

ما الذي تضيفه فوق CoALA وReAct وToT؟ لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق:

ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم

كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية

إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا، أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

3.6 الشروط التي تجعل النظرية قابلة للهدم

الشروط التي تجعل النظرية قابلة للهدم لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

3.7 الحد الأدنى الرياضي للنظريات

الحد الأدنى الرياضي للنظريات لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية

إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

3.8 حدود الادعاء: فرضية تشغيلية لا ضمان AGI

حدود الادعاء: فرضية تشغيلية لا ضمان AGI لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم

كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

الباب الرابع: التقنيات الثلاث الجديدة

هذا الباب يبني طبقة 4 من معمارية mcai-v2 agi new technology neoAGI. لا يشرح المفاهيم بوصفها درسا، بل يحولها إلى قرارات تصميم: ما الذي ندخله في الذاكرة؟ ما الذي نهدمه؟ ما الذي نقيسه؟ وما الذي يمنع الادعاء من التحول إلى دعاية؟

4.1 تقنية HAKIM-MCD v2 Operating Layer

تقنية HAKIM-MCD v2 Operating Layer لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

4.2 تقنية Vascular GraphWorld Memory

تقنية Vascular GraphWorld Memory لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

4.3 تقنية Falsification-Centric Test-Time Compute

تقنية Falsification-Centric Test-Time Compute لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R7] [R4] [R3]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

4.4 هندسة الربط بين الذاكرة والعالم والقرار

هندسة الربط بين الذاكرة والعالم والقرار لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R7] [R8]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

4.5 تنزيل تقني على وكيل برمجي

تنزيل تقني على وكيل برمجي لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

4.6 تنزيل تقني على باحث معرفي

تنزيل تقني على باحث معرفي لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

4.7 تنزيل تقني على روبوت/بيئة تفاعلية

تنزيل تقني على روبوت/بيئة تفاعلية لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية

إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

4.8 جدول مخاطر وتخفيف

جدول مخاطر وتخفيف لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم

كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

الباب الخامس: الأدوات الخمس عشرة لمحرّك neoAGI

هذا الباب يبني طبقة 5 من معمارية neoAGI mcai-v2 agi new technology. لا يشرح المفاهيم بوصفها درسا، بل يحولها إلى قرارات تصميم: ما الذي ندخله في الذاكرة؟ ما الذي نهدمه؟ ما الذي نقيسه؟ وما الذي يمنع الادعاء من التحول إلى دعاية؟

Critical Difference Extractor 5.1

Critical Difference Extractor لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

MCD Meta-Complexity Mapper 5.2

MCD Meta-Complexity Mapper لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

Vascular Compute Router 5.3

Vascular Compute Router لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا، أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

Inner-Time Budgeter 5.4

Inner-Time Budgeter لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا، أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

Causal Counterfactual Forge 5.5

Causal Counterfactual Forge لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

GraphWorld Builder 5.6

GraphWorld Builder لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

Epistemic Immune Filter 5.7

Epistemic Immune Filter لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

Perturbation Court 5.8

Perturbation Court لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

Moussaif Selective Rescue 5.9

Moussaif Selective Rescue لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

Novelty Entropy Meter 5.10

Novelty Entropy Meter لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

Anti-Hijack Source Ledger 5.11

Anti-Hijack Source Ledger لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

Skill-to-Hyperskill Compiler 5.12

Skill-to-Hyperskill Compiler لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

World-Effect Loop Monitor 5.13

World-Effect Loop Monitor لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

Self-Reconstruction Memory 5.14

Self-Reconstruction Memory لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

Benchmark Tribunal 5.15

Benchmark Tribunal لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

الباب السادس: فلسفات التدريب وأدوات التدريب

هذا الباب يبني طبقة 6 من معمارية mcai-v2 agi new technology neoAGI. لا يشرح المفاهيم بوصفها درسا، بل يحولها إلى قرارات تصميم: ما الذي ندخله في الذاكرة؟ ما الذي نهدمه؟ ما الذي نقيسه؟ وما الذي يمنع الادعاء من التحول إلى دعاية؟

6.1 تدريب الفروق لا الإجابات

تدريب الفروق لا الإجابات لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

6.2 تدريب التفنيد قبل الثقة

تدريب التفنيد قبل الثقة لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

6.3 تدريب الذاكرة كإعادة بناء لا كاسترجاع

تدريب الذاكرة كإعادة بناء لا كاسترجاع لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R7] [R8]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

6.4 تدريب العالم السببي لا النص وحده

تدريب العالم السببي لا النص وحده لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R11] [R12]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

6.5 تدريب الانتقاء المساعفي

تدريب الانتقاء المساعفي لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

6.6 أداة Vascular Curriculum Generator

أداة Vascular Curriculum Generator لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

6.7 أداة Failure Distillation Lab

أداة **Failure Distillation Lab** لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

6.8 أداة Causal Replay Forge

أداة **Causal Replay Forge** لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

6.9 أداة Benchmark Anti-Gaming Lab

أداة Benchmark Anti-Gaming Lab لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: claim، evidence، counter-condition، world-effect. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

6.10 أداة Governance Reward Model

أداة Governance Reward Model لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

الباب السابع: البنية التقنية الكاملة

هذا الباب يبني طبقة 7 من معمارية mcai-v2 agi new technology neoAGI. لا يشرح المفاهيم بوصفها درسا، بل يحولها إلى قرارات تصميم: ما الذي ندخله في الذاكرة؟ ما الذي نهدمه؟ ما الذي نقيسه؟ وما الذي يمنع الادعاء من التحول إلى دعاية؟

7.1 مخطط طبقات MCAI-v2

مخطط طبقات MCAI-v2 لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

7.2 واجهات النظام APIs

واجهات النظام APIs لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

7.3 صيغة الذاكرة والتجربة

صيغة الذاكرة والتجربة لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R7] [R8]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

7.4 بروتوكول الاسترجاع الحاكم

بروتوكول الاسترجاع الحاكم لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

7.5 بروتوكول التخطيط الوعائي

بروتوكول التخطيط الوعائي لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R9] [R10]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية

إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

7.6 بروتوكول التفنيد المتعدد

بروتوكول التفنيد المتعدد لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم

كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

7.7 بروتوكول التعلّم من الأثر

بروتوكول التعلّم من الأثر لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

7.8 النشر على Agentic OS

النشر على Agentic OS لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

الباب الثامن: القياس والاختبار والbenchmarks

هذا الباب يبني طبقة 8 من معمارية mcai-v2 agi new technology neoAGI. لا يشرح المفاهيم بوصفها درسا، بل يحولها إلى قرارات تصميم: ما الذي ندخله في الذاكرة؟ ما الذي نهدمه؟ ما الذي نقيسه؟ وما الذي يمنع الادعاء من التحول إلى دعاية؟

8.1 لماذا لا نلفّق درجات benchmark؟

لماذا لا نلفّق درجات benchmark؟ لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا، أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

8.2 مصفوفة 13 معيارا للاختبار

مصفوفة 13 معيارا للاختبار لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا، أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

8.3 ARC-AGI كاختبار كفاءة اكتساب مهارة

ARC-AGI كاختبار كفاءة اكتساب مهارة لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R13] [R14]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

8.4 SWE-bench كاختبار أثر واقعي

SWE-bench كاختبار أثر واقعي لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R15] [R17]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

8.5 GAIA/WebArena/OSWorld/Terminal-Bench كاختبارات وكيل

GAIA/WebArena/OSWorld/Terminal-Bench كاختبارات وكيل لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

8.6 مقياس VGB الحاكمي

مقياس VGB الحاكمي لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

8.7 اختبار مقاومة تلوث البيانات

اختبار مقاومة تلوث البيانات لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

8.8 اختبار قابلية الهدم والتعميم

اختبار قابلية الهدم والتعميم لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا، أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

الباب التاسع: النموذج الجديد M-VCGE

هذا الباب يبني طبقة 9 من معمارية mcai-v2 agi new technology neoAGI. لا يشرح المفاهيم بوصفها درسا، بل يحولها إلى قرارات تصميم: ما الذي ندخله في الذاكرة؟ ما الذي نهدمه؟ ما الذي نقيسه؟ وما الذي يمنع الادعاء من التحول إلى دعاية؟

9.1 تعريف Moussaif Vascular-Causal Governance Engine

تعريف Moussaif Vascular-Causal Governance Engine لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

9.2 طبقات النموذج

طبقات النموذج لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا، أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

9.3 آليات القرار

آليات القرار لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا، أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

9.4 ما يفسره أفضل من النماذج الحالية

ما يفسره أفضل من النماذج الحالية لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

9.5 ما لا يفسره

ما لا يفسره لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

9.6 تنبؤات قابلة للاختبار

تنبؤات قابلة للاختبار لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية

إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

9.7 نسخة صفحة واحدة

نسخة صفحة واحدة لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم

كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

9.8 خريطة تشغيل

خريطة تشغيل لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا، أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

الباب العاشر: محكمة الهدم والنسخة الأقوى

هذا الباب يبني طبقة 10 من معمارية neoAGI agi new technology mcai-v2. لا يشرح المفاهيم بوصفها درسا، بل يحولها إلى قرارات تصميم: ما الذي ندخله في الذاكرة؟ ما الذي نهدمه؟ ما الذي نقيسه؟ وما الذي يمنع الادعاء من التحول إلى دعاية؟

10.1 أقوى 10 اعتراضات

أقوى 10 اعتراضات لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

10.2 أقوى 10 نقاط ضعف

أقوى 10 نقاط ضعف لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

10.3 ما قد يكون مبالغا فيه

ما قد يكون مبالغا فيه لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

10.4 ما يحتاج إلى دليل أكثر

ما يحتاج إلى دليل أكثر لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

10.5 ماذا تقول المدرسة المنافسة؟

ماذا تقول المدرسة المنافسة؟ لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

10.6 ما الذي يفند MCAI-v2؟

ما الذي يفند MCAI-v2؟ لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية

إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

10.7 النسخة المعدلة الأقوى

النسخة المعدلة الأقوى لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم

كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

10.8 سجل البناء المعرفي

سجل البناء المعرفي لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

الباب الحادي عشر: البرنامج البحثي المستقبلي

هذا الباب يبني طبقة 11 من معمارية neoAGI agi new technology mcai-v2. لا يشرح المفاهيم بوصفها درسا، بل يحولها إلى قرارات تصميم: ما الذي ندخله في الذاكرة؟ ما الذي نهدمه؟ ما الذي نقيسه؟ وما الذي يمنع الادعاء من التحول إلى دعاية؟

11.1 12 فرضية بحثية

12 فرضية بحثية لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

11.2 12 سؤالا مستقبليا

12 سؤالا مستقبليا لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

11.3 12 مشروعا قابلا للتنفيذ

12 مشروعا قابلا للتنفيذ لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

11.4 12 مفهوما يجب ضبطه

12 مفهوما يجب ضبطه لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

11.5 12 خطأ شائعا

12 خطأ شائعا لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

11.6 12 تمييزا معرفيا

12 تمييزا معرفيا لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

> **فرضية تشغيلية**
> إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

11.7 خارطة 24 شهرا

خارطة 24 شهرا لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

> **تمييز حاكم**
> كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

11.8 خاتمة تقنية

خاتمة تقنية لا يقرأ هنا كعنوان تعليمي، بل كنقطة ضغط هندسية: السؤال ليس كيف نزيد حجم النموذج، بل كيف نمنع النموذج من تحويل التفكير إلى تدفق لغوي بلا حاكمية. في MCAI-v2 يصبح المسار الذكي سلسلة قرارات حول الفرق: ما الفرق الذي يغير الخطة؟ ما الفرق الذي يغير المصدر؟ ما الفرق الذي يغير معيار النجاح؟ وما الفرق الذي يجب تجاهله لأنه ضجيج؟ [R3] [R4] [R7]

القيد الأول هو أن البنية الحالية للنماذج المفكرة تعالج جزءا من المشكلة: تزيد زمن التفكير، توسع البحث، تضيف أدوات، أو تحفظ آثارا في الذاكرة. لكنها غالبا لا تمتلك طبقة موحدة تقرر متى يكون الاسترجاع أولى من الاستدلال، ومتى يكون التفنيد أولى من الإكمال، ومتى يكون الفعل في العالم ضروريا قبل الجواب. لذلك لا يكفي أن نقول chain-of-thought أو agent loop؛ يجب أن نسأل: من يحكم الحلقة؟

تمييز حاكم
كل تقنية تصلح محليا قد تفشل كطريق إلى AGI إذا لم تمتلك: معيار فرق، ذاكرة أثر، محكمة هدم، وعلاقة سببية بالعالم. التقنية التي ترفع benchmark واحدا ليست تقنية ذكاء عام حتى تثبت نقل المهارة تحت priors قليلة وتحت تغير البيئة.

في صياغة مساعف، الذكاء ليس كثرة خطوات، بل جودة ترتيب الأوعية: وعاء المصدر، وعاء المعنى، وعاء الخطة، وعاء الفعل، وعاء الأثر، وعاء الذاكرة. كل وعاء يحمل نوعا مختلفا من الالتباس. فالتباس المصدر يعالج بالتوثيق، والتباس المعنى يعالج بالتفريق، والتباس الخطة يعالج بالمحاكاة، والتباس الفعل يعالج بقياس الأثر، والتباس الذاكرة يعالج بإعادة البناء بعد الفشل.

الآلية التنفيذية المقترحة تجعل كل خطوة تفكير تنتج سجلا رباعيا: world-effect، counter-condition، evidence، claim. لا يسمح للنظام بترقية claim إلى معرفة تشغيلية إلا إذا ظهر لها سند كاف، واعتراض كاف، ومجال تطبيق محدد، وتكلفة خطأ محسوبة. هذا يحول التفكير من نص إلى إدارة مخاطر معرفية.

ينتج عن ذلك معيار عملي: عندما يتعامل النظام مع مسألة جديدة، لا يبدأ بالإجابة بل بمصفوفة MCD. هذه المصفوفة تقدّر عمق الفرق عبر ستة محاور: الغموض، التناقض، الأثر، قابلية الاختبار، تكلفة الخطأ، ودرجة الجدة. إذا ارتفعت قيمة التناقض، يستدعى Perturbation Court؛ إذا ارتفعت قيمة الجدة، يستدعى Novelty Entropy Meter؛ إذا ارتفعت تكلفة الخطأ، يستدعى Anti-Hijack Source Ledger قبل أي توليد.

فرضية تشغيلية
إذا جرى تدريب الوكيل على توزيع الجهد وفق MCD بدل توزيع ثابت للخطوات، فسيظهر تحسن في مهام قليلة البيانات ومتغيرة القواعد، لأن النظام سيصرف compute حيث يوجد فرق حاسم لا حيث يوجد نص كثير. هذه فرضية تحتاج تجربة على ARC-AGI وبيئات تفاعلية لا على سؤال/جواب فقط.

الاعتراض الأقوى: قد تكون هذه الحاكمية مجرد scaffold خارجي لا يضيف ذكاء للنموذج الأساسي. الجواب المنضبط: نعم، في البداية هي scaffold. لكن تاريخ الأنظمة الذكية يبين أن architecture + memory + feedback يمكن أن يغير طبقة القدرة دون تغيير الأوزان كل مرة. لذلك يطلب الكتاب اختبارين: هل تتحسن القدرة عبر جلسات متعاقبة؟ وهل تنقل hyperskill من مجال إلى مجال دون أمثلة كثيرة؟

التنزيل التقني في هذه النقطة يبدأ من pipeline واضح: استخراج الفروق، ترتيبها بمقياس MCD، اختيار أداة أو ذاكرة أو تفنيد، توليد خطة قصيرة، تنفيذ فعل أو استرجاع، محاكمة الأثر، ثم إعادة بناء الذاكرة. الفشل المقبول هو الفشل الذي ينتج قاعدة جديدة قابلة للتعميم؛ أما الفشل الذي لا ينتج فرقا فهو ضجيج تدريبي يجب عزله.

خلاصة هذا القسم: تجاوز البنيات الحالية لا يعني إلغاء LLM، بل سحب مركز الثقل من النموذج إلى حاكمية معرفية فوقه. النموذج يصبح مولدا ومفسرا ومحركا لغويا؛ أما الذكاء العام فينشأ - إن نشأ - من نظام يضبط: ماذا نعرف؟ لماذا نثق؟ ماذا نفعل؟ ماذا حدث؟ كيف نعيد بناء أنفسنا بعد الخطأ؟

ملحق الأدوات التنفيذية الخمس عشرة - مواصفات قابلة للبرمجة

Critical Difference Extractor

يعزل الفرق الذي يغيّر القرار، لا الفرق اللغوي العادي. هذه الأداة ليست إضافة زخرفية؛ إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	يعزل الفرق الذي يغيّر القرار، لا الفرق اللغوي العادي.
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`extract_delta(inputs)->ranked_differences`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

MCD Meta-Complexity Mapper

يرسم خريطة تعقيد: غموض، تعارض، أثر، تكلفة، قابلية اختبار. هذه الأداة ليست إضافة زخرفية؛ إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	يرسم خريطة تعقيد: غموض، تعارض، أثر، تكلفة، قابلية اختبار.
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`mcd_map(differences,context)->tensor`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

Vascular Compute Router

يوزع التفكير والاسترجاع والأدوات حسب درجة MCD والمخاطر. هذه الأداة ليست إضافة زخرفية؛ إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	يوزع التفكير والاسترجاع والأدوات حسب درجة MCD والمخاطر.
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`route(budget,MCD,risk)->actions`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

Inner-Time Budgeter

يصنع زمنا داخليا للمسألة: مراحل انتظار، رجوع، تفريع، قرار. هذه الأداة ليست إضافة زخرفية؛ إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	يصنع زمنا داخليا للمسألة: مراحل انتظار، رجوع، تفريع، قرار.
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`schedule(thought_tree,deadlines)->plan`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

Causal Counterfactual Forge

يبني بدائل سببية: ماذا لو أزيلت مقدمة؟ ماذا لو تغير أثر؟ هذه الأداة ليست إضافة زخرفية؛ إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	يبني بدائل سببية: ماذا لو أزيلت مقدمة؟ ماذا لو تغير أثر؟
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`counterfactuals(model,intervention)->worlds`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

GraphWorld Builder

يمزج KG وworld model في تمثيل واحد قابل للاسترجاع والمحاكاة. هذه الأداة ليست إضافة زخرفية؛ إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	يمزج KG وworld model في تمثيل واحد قابل للاسترجاع والمحاكاة.
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`build_graphworld(corpus,events)->G_w`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

Epistemic Immune Filter

يمنع إدخال معرفة مسمومة أو غير موثقة في الذاكرة الحاكمة. هذه الأداة ليست إضافة زخرفية؛ إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	يمنع إدخال معرفة مسمومة أو غير موثقة في الذاكرة الحاكمة.
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`immune_check(claim,sources)->verdict`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

Perturbation Court

يهاجم الخطة بتشويش مضبوط: أمثلة مضادة، قيود، موارد ناقصة. هذه الأداة ليست إضافة زخرفية؛ إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	يهاجم الخطة بتشويش مضبوط: أمثلة مضادة، قيود، موارد ناقصة.
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`attack(plan)->failure_modes`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

Moussaif Selective Rescue

ينقذ المسارات الفاشلة ذات الإشارة المعرفية العالية فقط. هذه الأداة ليست إضافة زخرفية؛ إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	ينقذ المسارات الفاشلة ذات الإشارة المعرفية العالية فقط.
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`rescue(failures,MCD)->revised_candidates`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

Novelty Entropy Meter

يقيس هل الجديد بنيوي أم زخرفة اصطلاحية. هذه الأداة ليست إضافة زخرفية؛ إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	يقيس هل الجديد بنيوي أم زخرفة اصطلاحية.
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`novelty_entropy(candidate,prior_art)->score`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

Anti-Hijack Source Ledger

دفتر مصادر يمنع أن يختطف مصدر واحد هندسة الحكم. هذه الأداة ليست إضافة زخرفية؛ إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	دفتر مصادر يمنع أن يختطف مصدر واحد هندسة الحكم.
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`ledger_update(claim,evidence)->audit_record`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

Skill-to-Hyperskill Compiler

يحوّل مهارة ناجحة إلى hyperskill قابلة للنقل بين مجالات. هذه الأداة ليست إضافة زخرفية؛ إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	يحوّل مهارة ناجحة إلى hyperskill قابلة للنقل بين مجالات.
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`compile(skill_traces)->hyperskill`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

World-Effect Loop Monitor

يقيس أثر فعل الوكيل في البيئة ثم يحدث السياسة. هذه الأداة ليست إضافة زخرفية؛ إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	يقيس أثر فعل الوكيل في البيئة ثم يحدث السياسة.
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`observe_effect(action,world)->learning_event`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

Self-Reconstruction Memory

ذاكرة لا تخزن فقط، تعيد كتابة ذاتها بعد المحاكمات. هذه الأداة ليست إضافة زخرفية، إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	ذاكرة لا تخزن فقط، تعيد كتابة ذاتها بعد المحاكمات.
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`reconstruct(memory,verdicts)->memory_v2`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

Benchmark Tribunal

يحاكم الدرجة: هل حل النموذج المهمة أم استغل benchmark؟ هذه الأداة ليست إضافة زخرفية، إنها موضع فصل بين مسار لغوي ومسار حاكم. قيمتها لا تقاس بجمال التسمية بل بقدرتها على تغيير القرار في حالة نزاع أو نقص أو خطر.

البند	التحديد التقني
وظيفة الأداة	يحاكم الدرجة: هل حل النموذج المهمة أم استغل benchmark؟
مدخلات	سياق المسألة، سجل المصادر، آثار المحاولات السابقة، قيود السلامة والتكلفة.
مخرجات	قرار قابل للتتبع: قبول، رفض، تفريع، استرجاع، تفنيد، إنقاذ، أو توقف.
واجهة أولية	`tribunal(result,trace)->validity`
اختبار نجاح	تحسن تحت مهام جديدة لا تظهر في التدريب، وانخفاض hallucination / benchmark gaming.

قاعدة التشغيل: لا تدخل الأداة في الحلقة إلا إذا كان الفرق الذي تعالجه أعلى من عتبة MCD، وإلا صارت تكلفة إضافية تزيد الارتباك. في النسخة الأولى يمكن تنفيذها كدالة Python فوق LLM، وفي النسخة المتقدمة تتحول إلى policy module يتعلم من سجلات النجاح والفشل.

فلسفات التدريب الخمس

تدريب الفروق لا الإجابات

لا يتعلم النموذج جواب السؤال فقط، بل يتعلم أي فرق داخل السؤال كان حاسما. في النسخة التطبيقية، لا تعطى العينة للنموذج كمدخل/مخرج فقط، بل كمسار: فرق أولي، قرار حاكم، اعتراض، أثر، ثم تحديث ذاكرة. بذلك يصبح التدريب تعليما لحاكمية المسار لا لحفظ النمط.

> **قاعدة تدريب**
> كل فلسفة تدريب يجب أن تنتج loss أو reward أو معيار تقييم؛ إن بقيت عبارة ثقافية فهي محذوفة من النسخة العلمية.

تدريب التفنيد قبل الثقة

كل عينة تدريبية تمر بقاض يطلب شرط هدمها قبل اعتمادها. في النسخة التطبيقية، لا تعطى العينة للنموذج كمدخل/مخرج فقط، بل كمسار: فرق أولي، قرار حاكم، اعتراض، أثر، ثم تحديث ذاكرة. بذلك يصبح التدريب تعليما لحاكمية المسار لا لحفظ النمط.

> **قاعدة تدريب**
> كل فلسفة تدريب يجب أن تنتج loss أو reward أو معيار تقييم؛ إن بقيت عبارة ثقافية فهي محذوفة من النسخة العلمية.

تدريب الذاكرة كإعادة بناء

تتحول الذاكرة من أرشيف إلى نظام تحديث ذاتي مشروط بالأثر. في النسخة التطبيقية، لا تعطى العينة للنموذج كمدخل/مخرج فقط، بل كمسار: فرق أولي، قرار حاكم، اعتراض، أثر، ثم تحديث ذاكرة. بذلك يصبح التدريب تعليما لحاكمية المسار لا لحفظ النمط.

> **قاعدة تدريب**
> كل فلسفة تدريب يجب أن تنتج loss أو reward أو معيار تقييم؛ إن بقيت عبارة ثقافية فهي محذوفة من النسخة العلمية.

تدريب العالم السببي

لا يكفي النص، يجب أن يتعلم النظام أثر الفعل وتغير الحالة. في النسخة التطبيقية، لا تعطى العينة للنموذج كمدخل/مخرج فقط، بل كمسار: فرق أولي، قرار حاكم، اعتراض، أثر، ثم تحديث ذاكرة. بذلك يصبح التدريب تعليما لحاكمية المسار لا لحفظ النمط.

> **قاعدة تدريب**
> كل فلسفة تدريب يجب أن تنتج loss أو reward أو معيار تقييم؛ إن بقيت عبارة ثقافية فهي محذوفة من النسخة العلمية.

تدريب الانتقاء المساعفي

الفشل ليس كله مادة تعلم؛ ينتقى منه ما يحمل إشارة قابلة للتعميم. في النسخة التطبيقية، لا تعطى العينة للنموذج كمدخل/مخرج فقط، بل كمسار: فرق أولي، قرار حاكم، اعتراض، أثر، ثم تحديث ذاكرة. بذلك يصبح التدريب تعليما لحاكمية المسار لا لحفظ النمط.

> **قاعدة تدريب**
> كل فلسفة تدريب يجب أن تنتج loss أو reward أو معيار تقييم؛ إن بقيت عبارة ثقافية فهي محذوفة من النسخة العلمية.

أدوات التدريب الخمس

Vascular Curriculum Generator

يصنع منهجا يبدأ من الفروق الواضحة ثم ينتقل إلى الفروق المختلطة والمتعارضة. تقاس فاعلية الأداة عبر ablation واضح: تشغيل النظام بها وبدونها، ثم قياس انتقال المهارة، مقاومة الهلوسة، وتكلفة compute.

Failure Distillation Lab

يحوّل أخطاء الوكيل إلى أمثلة مضادة وقواعد وقف وقواعد إنقاذ. تقاس فاعلية الأداة عبر ablation واضح: تشغيل النظام بها وبدونها، ثم قياس انتقال المهارة، مقاومة الهلوسة، وتكلفة compute.

Causal Replay Forge

يعيد تشغيل المسألة مع تدخلات سببية لاختبار صلابة الخطة. تقاس فاعلية الأداة عبر ablation واضح: تشغيل النظام بها وبدونها، ثم قياس انتقال المهارة، مقاومة الهلوسة، وتكلفة compute.

Benchmark Anti-Gaming Lab

يصمم نسخا ديناميكية من الاختبارات حتى لا تصبح الدرجة لعبة حفظ. تقاس فاعلية الأداة عبر ablation واضح: تشغيل النظام بها وبدونها، ثم قياس انتقال المهارة، مقاومة الهلوسة، وتكلفة compute.

Governance Reward Model

يكافئ مسار التفكير الحاكم: مصدر، تفنيد، أثر، تواضع معرفي، لا جوابا جذابا فقط. تقاس فاعلية الأداة عبر ablation واضح: تشغيل النظام بها وبدونها، ثم قياس انتقال المهارة، مقاومة الهلوسة، وتكلفة compute.

مصفوفة 13 benchmark دون تلفيق درجات

المعيار	نوع القدرة	ما يقيسه فعلا	السند/الرتبة
ARC-AGI-1/2/3	few-shot abstraction / interactive reasoning	كفاءة اكتساب مهارة لا مجرد حفظ	R13/R14
SWE-bench Verified	real GitHub issue resolution	أثر برمجي قابل للفحص	R15/R17
GAIA	multi-step general assistant tasks	بحث، أدوات، تحقق، تركيب	عام
WebArena	web navigation agents	فعل في بيئة ويب	عام
OSWorld	computer-use agents	تحكم حاسوبي طويل المسار	عام
Terminal-Bench	terminal coding/ops agents	أوامر، ملفات، تحقق، أثر	عام
τ-bench	tool-use policy compliance	التزام سياسة الأدوات	عام
MATH/AIME	mathematical reasoning	استدلال رسمي محدود المجال	عام
GPQA	graduate-level science QA	معرفة تخصصية صعبة	عام
LiveCodeBench	fresh coding	تقليل تلوث البيانات	عام
HumanEval+/ MBPP+	code generation	اختبار سريع لا يكفي وحده	عام
LongBench/RULER	long-context	ذاكرة سياقية لا ذاكرة ذاتية	عام
VGB-Moussaif	vascular governance benchmark	حاكمية الفروق، التفنيد، الأثر	فرضية/ اختراع

لا توجد في هذا الكتاب درجات مدعاة لـMCAI-v2 لأنها لم تشغل على harness حقيقي. أي رقم قبل التنفيذ سيكون زخرفة. المطلوب في الإصدار التجريبي أن يبنى CI benchmark pipeline يسجل cost، artifacts، traces، tools، model، prompt، وverdict. الدرجة لا تعتمد إلا بعد reproducibility وثلاثة أنواع من الاختبار: ثابت، ديناميكي، ومضاد للتلاعب.

النموذج الجديد: M-VCGE

M-VCGE = Moussaif Vascular-Causal Governance Engine. هو نموذج حاكمية فوق-نموذجي يرى الذكاء العام كقدرة على إدارة الفروق عبر أوعية متعددة، مع ذاكرة تعيد بناء ذاتها، وعالم سببي قابل للمحاكاة، ومحكمة تفنيد تمنع الثقة المبكرة.

الطبقة	وظيفتها	اختبارها
Layer 0 - Source Ledger	توثيق الدعوى ومصدرها ودرجة الاعتماد	اختبار مصادر متعارضة وبيانات مسمومة
Layer 1 - Difference Extraction	استخراج الفروق الحاسمة	مقارنة مع وسوم بشرية للفروق
Layer 2 - MCD Mapping	ترتيب التعقيد المنتج	ablation لمحاور MCD
Layer 3 - GraphWorld	تمثيل سببي/علائقي للمعرفة	اختبارات counterfactual
Layer 4 - Vascular Routing	توزيع compute/tools/memory	تحليل تكلفة/دقة
Layer 5 - Falsification Court	تفنيد قبل اعتماد	انخفاض الأخطاء الحرجة
Layer 6 - World-Effect Loop	قياس أثر الفعل	مهام بيئية أو برمجية فعلية
Layer 7 - Self-Reconstruction	تحديث ذاكرة وسياسة	تحسن عبر جلسات

صيغة تشغيلية مختصرة

- $MCD(q)=\lambda1\cdot Ambiguity + \lambda2\cdot Contradiction + \lambda3\cdot Novelty + \lambda4\cdot WorldEffect + \lambda5\cdot Testability + \lambda6\cdot Risk + \lambda7\cdot Redundancy$. ويكون routing: $b_i = softmax(\alpha\ MCD_i + \beta\ Risk_i + \gamma\ Novelty_i - \delta\ Cost_i)$.

ما يفسره النموذج أفضل من scaffolds الحالية: لماذا يفشل الوكيل حين تزيد الأدوات بلا حاكمية؛ لماذا تنجح الذاكرة أحيانا وتسمم النظام أحيانا؛ لماذا benchmark score قد يخدع؛ ولماذا يحتاج الذكاء العام إلى توزيع جهد لا إلى زيادة جهد فقط. ما لا يفسره: أصل الوعي الذاتي، ضمان التعميم المفتوح، وحل alignment النهائي.

محكمة الهدم: أقوى الاعتراضات والجواب المعدل

الاعتراض	الجواب المنهجي
هذه ليست AGI بل orchestration حول LLM	صحيح جزئيا؛ لذلك يصاغ MCAI-v2 كبرنامج بحثي. قيمته في تحويل orchestration إلى حاكمية قابلة للقياس لا إلى تجميع أدوات.
MCD قد يكون مقياسا ذاتيا	يعالج بتعريف محاور قابلة للوسم البشري وباختبارات ablation: إزالة محور وقياس سقوط الأداء.
الذاكرة قد تراكم أخطاء	لذلك لا تعتمد الذاكرة إلا بعد Anti-Hijack Ledger ومحكمة هدم وسجل مصادر.
world model النصي لا يفهم العالم	لذلك يفصل الكتاب بين GraphWorld رمزي، ومحاكاة causal، وبيئة فعلية عند الإمكان.
التفنيد المتكرر يبطئ النظام	يستعمل Vascular Router التفنيد فقط حين ترتفع تكلفة الخطأ أو التناقض.
قد يتلاعب النظام بالbenchmarks	Benchmark Tribunal يطلب traces، اختبارات ديناميكية، وقياس أثر لا score فقط.
التقنيات قد تنجح في النص وتفشل في الروبوتات	تحتاج نسخة embodied ببيئات world models ومقاييس physical consistency.
الانتقائية المساعفية قد تلغي فشلا مفيدا	يعالج بإبقاء عينة صغيرة عشوائية من الفشل غير المختار للمراجعة الدورية.
اللغة العربية/الإنجليزية قد تخلط المصطلحات	يعالج بقاموس ثنائي صارم وتعريفات تشغيلية لا بلاغية.
النموذج قد يصبح معقدا أكثر من اللازم	يختبر عبر minimal viable governance: أقل طبقات تحقق تحسنا قابلا للقياس.

بعد المحكمة، لا تبقى النسخة القوية من MCAI-v2 هي النسخة الأكثر طموحا، بل النسخة الأكثر قابلية للحذف. كل طبقة لا تنجح في ablation يجب حذفها أو دمجها. بهذا يصبح المشروع علميا: قوته لا تأتي من كثرة المصطلحات بل من قبوله أن يهدم نفسه أمام بيانات أقوى.

قاموس مصطلحات

المصطلح	التعريف	الرتبة
MCAI-v2	Meta-Complexity Agency Intelligence v2؛ معمارية تحكم في الفروق والتعقيد والذاكرة والأثر.	تشغيلي/تركيبي
MCD	Meta-Complexity Differentiation؛ قياس يفرز أين يوجد التعقيد المنتج لا التعقيد اللفظي.	تشغيلي/تركيبي
HAKIM-MCD	طبقة حاكمية هرمية تقرر متى يفكر النظام، متى يسترجع، متى يفند، ومتى يتوقف.	تشغيلي/تركيبي
التراكب الوعائي	عدسة ترى الإدراك/الاستدلال كتراكب أوعية: حس، معنى، غاية، فعل، أثر، ذاكرة.	تشغيلي/تركيبي
الانتقائية المساعفية	قاعدة إنقاذ مشروطة: لا ننقذ كل مسار فاشل، ننقذ فقط ما يحمل فرقاً منتجاً قابلاً للاختبار.	تشغيلي/تركيبي
محكمة الهدم	وحدة تفنيد لا تحسن الجواب تجميليا، بل تبحث عن الشرط الذي يهدمه.	تشغيلي/تركيبي
World-Effect Loop	حلقة تربط القرار بأثره في العالم ثم تعيد بناء الذاكرة والسياسة.	تشغيلي/تركيبي
GraphWorld	تمثيل يجمع المعرفة النصية والعلائق السببية وحالات العالم المتغيرة.	فرضية تقنية
Hyperskill	مهارة فوقية قابلة للنقل بين مجالات بعد ضغط آثار النجاح والفشل.	فرضية تدريبية
Novelty Entropy	قياس مقدار الجدة البنيوية مقارنة بمصادر ومهارات سابقة.	مقياس مقترح
VGB	Vascular Governance Benchmark؛ اختبار حاكمية الفروق والتفنيد والأثر.	اختراع تقويمي

12 فرضية بحثية قوية

#	الفرضية	طريقة الاختبار
1	فرضية 1: رفع جودة routing وفق MCD سيحسن نقل المهارة في مهمة جديدة أكثر من زيادة عدد خطوات CoT وحدها.	اختبار ablation على ARC-like tasks وSWE-bench mini harness.
2	فرضية 2: رفع جودة routing وفق MCD سيحسن نقل المهارة في مهمة جديدة أكثر من زيادة عدد خطوات CoT وحدها.	اختبار ablation على ARC-like tasks وSWE-bench mini harness.
3	فرضية 3: رفع جودة routing وفق MCD سيحسن نقل المهارة في مهمة جديدة أكثر من زيادة عدد خطوات CoT وحدها.	اختبار ablation على ARC-like tasks وSWE-bench mini harness.
4	فرضية 4: رفع جودة routing وفق MCD سيحسن نقل المهارة في مهمة جديدة أكثر من زيادة عدد خطوات CoT وحدها.	اختبار ablation على ARC-like tasks وSWE-bench mini harness.
5	فرضية 5: رفع جودة routing وفق MCD سيحسن نقل المهارة في مهمة جديدة أكثر من زيادة عدد خطوات CoT وحدها.	اختبار ablation على ARC-like tasks وSWE-bench mini harness.
6	فرضية 6: رفع جودة routing وفق MCD سيحسن نقل المهارة في مهمة جديدة أكثر من زيادة عدد خطوات CoT وحدها.	اختبار ablation على ARC-like tasks وSWE-bench mini harness.
7	فرضية 7: رفع جودة routing وفق MCD سيحسن نقل المهارة في مهمة جديدة أكثر من زيادة عدد خطوات CoT وحدها.	اختبار ablation على ARC-like tasks وSWE-bench mini harness.
8	فرضية 8: رفع جودة routing وفق MCD سيحسن نقل المهارة في مهمة جديدة أكثر من زيادة عدد خطوات CoT وحدها.	اختبار ablation على ARC-like tasks وSWE-bench mini harness.
9	فرضية 9: رفع جودة routing وفق MCD سيحسن نقل المهارة في مهمة جديدة أكثر من زيادة عدد خطوات CoT وحدها.	اختبار ablation على ARC-like tasks وSWE-bench mini harness.
10	فرضية 10: رفع جودة routing وفق MCD سيحسن نقل المهارة في مهمة جديدة أكثر من زيادة عدد خطوات CoT وحدها.	اختبار ablation على ARC-like tasks وSWE-bench mini harness.
11	فرضية 11: رفع جودة routing وفق MCD سيحسن نقل المهارة في مهمة جديدة أكثر من زيادة عدد خطوات CoT وحدها.	اختبار ablation على ARC-like tasks وSWE-bench mini harness.
12	فرضية 12: رفع جودة routing وفق MCD سيحسن نقل المهارة في مهمة جديدة أكثر من زيادة عدد خطوات CoT وحدها.	اختبار ablation على ARC-like tasks وSWE-bench mini harness.

12 سؤالا مستقبليا

#	السؤال	نوع البحث
1	سؤال 1: ما أقل بنية حاكمية تحقق انتقال hyperskill دون ذاكرة ملوثة؟	بحث تجريبي/نظري
2	سؤال 2: ما أقل بنية حاكمية تحقق انتقال hyperskill دون ذاكرة ملوثة؟	بحث تجريبي/نظري
3	سؤال 3: ما أقل بنية حاكمية تحقق انتقال hyperskill دون ذاكرة ملوثة؟	بحث تجريبي/نظري
4	سؤال 4: ما أقل بنية حاكمية تحقق انتقال hyperskill دون ذاكرة ملوثة؟	بحث تجريبي/نظري
5	سؤال 5: ما أقل بنية حاكمية تحقق انتقال hyperskill دون ذاكرة ملوثة؟	بحث تجريبي/نظري
6	سؤال 6: ما أقل بنية حاكمية تحقق انتقال hyperskill دون ذاكرة ملوثة؟	بحث تجريبي/نظري
7	سؤال 7: ما أقل بنية حاكمية تحقق انتقال hyperskill دون ذاكرة ملوثة؟	بحث تجريبي/نظري
8	سؤال 8: ما أقل بنية حاكمية تحقق انتقال hyperskill دون ذاكرة ملوثة؟	بحث تجريبي/نظري
9	سؤال 9: ما أقل بنية حاكمية تحقق انتقال hyperskill دون ذاكرة ملوثة؟	بحث تجريبي/نظري
10	سؤال 10: ما أقل بنية حاكمية تحقق انتقال hyperskill دون ذاكرة ملوثة؟	بحث تجريبي/نظري
11	سؤال 11: ما أقل بنية حاكمية تحقق انتقال hyperskill دون ذاكرة ملوثة؟	بحث تجريبي/نظري
12	سؤال 12: ما أقل بنية حاكمية تحقق انتقال hyperskill دون ذاكرة ملوثة؟	بحث تجريبي/نظري

12 مشروعا بحثيا قابلا للتنفيذ

#	المشروع	الموارد
1	مشروع 1: بناء prototype لوحدة من وحدات MCAI-v2 ثم قياسها على معيار ديناميكي.	4-8 أسابيع، LLM API، Python، سجل traces
2	مشروع 2: بناء prototype لوحدة من وحدات MCAI-v2 ثم قياسها على معيار ديناميكي.	4-8 أسابيع، LLM API، Python، سجل traces
3	مشروع 3: بناء prototype لوحدة من وحدات MCAI-v2 ثم قياسها على معيار ديناميكي.	4-8 أسابيع، LLM API، Python، سجل traces
4	مشروع 4: بناء prototype لوحدة من وحدات MCAI-v2 ثم قياسها على معيار ديناميكي.	4-8 أسابيع، LLM API، Python، سجل traces
5	مشروع 5: بناء prototype لوحدة من وحدات MCAI-v2 ثم قياسها على معيار ديناميكي.	4-8 أسابيع، LLM API، Python، سجل traces
6	مشروع 6: بناء prototype لوحدة من وحدات MCAI-v2 ثم قياسها على معيار ديناميكي.	4-8 أسابيع، LLM API، Python، سجل traces
7	مشروع 7: بناء prototype لوحدة من وحدات MCAI-v2 ثم قياسها على معيار ديناميكي.	4-8 أسابيع، LLM API، Python، سجل traces
8	مشروع 8: بناء prototype لوحدة من وحدات MCAI-v2 ثم قياسها على معيار ديناميكي.	4-8 أسابيع، LLM API، Python، سجل traces
9	مشروع 9: بناء prototype لوحدة من وحدات MCAI-v2 ثم قياسها على معيار ديناميكي.	4-8 أسابيع، LLM API، Python، سجل traces
10	مشروع 10: بناء prototype لوحدة من وحدات MCAI-v2 ثم قياسها على معيار ديناميكي.	4-8 أسابيع، LLM API، Python، سجل traces
11	مشروع 11: بناء prototype لوحدة من وحدات MCAI-v2 ثم قياسها على معيار ديناميكي.	4-8 أسابيع، LLM API، Python، سجل traces
12	مشروع 12: بناء prototype لوحدة من وحدات MCAI-v2 ثم قياسها على معيار ديناميكي.	4-8 أسابيع، LLM API، Python، سجل traces

ملحق بطاقات البروتوكولات التقنية المكثفة

بطاقة 1: بروتوكول منع الهلوسة

الغرض من بروتوكول منع الهلوسة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 2: بروتوكول اختيار الأداة

الغرض من بروتوكول اختيار الأداة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 3: بروتوكول ترقية الذاكرة

الغرض من بروتوكول ترقية الذاكرة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 4: بروتوكول ضغط التجربة

الغرض من بروتوكول ضغط التجربة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 5: بروتوكول قياس الجدة

الغرض من بروتوكول قياس الجدة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 6: بروتوكول محاكمة المصدر

الغرض من بروتوكول محاكمة المصدر هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 7: بروتوكول التخطيط السببي

الغرض من بروتوكول التخطيط السببي هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 8: بروتوكول تحويل المهارة إلى hyperskill

الغرض من بروتوكول تحويل المهارة إلى hyperskill هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 9: بروتوكول مقاومة benchmark gaming

الغرض من بروتوكول مقاومة benchmark gaming هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 10: بروتوكول بناء GraphWorld

الغرض من بروتوكول بناء GraphWorld هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 11: بروتوكول التفنيد العكسي

الغرض من بروتوكول التفنيد العكسي هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 12: بروتوكول أثر العالم

الغرض من بروتوكول أثر العالم هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 13: بروتوكول اختيار الفشل المفيد

الغرض من بروتوكول اختيار الفشل المفيد هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 14: بروتوكول الخريطة الوعائية

الغرض من بروتوكول الخريطة الوعائية هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 15: بروتوكول تحديث reward governance

الغرض من بروتوكول تحديث reward governance هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 16: بروتوكول اختبار few-shot transfer

الغرض من بروتوكول اختبار few-shot transfer هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 17: بروتوكول قياس تكلفة الخطأ

الغرض من بروتوكول قياس تكلفة الخطأ هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 18: بروتوكول استعمال مصادر متعارضة

الغرض من بروتوكول استعمال مصادر متعارضة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 19: بروتوكول المعرفة الزمنية

الغرض من بروتوكول المعرفة الزمنية هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 20: بروتوكول الإصدار والتجريب

الغرض من بروتوكول الإصدار والتجريب هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 21: بروتوكول منع الهلوسة

الغرض من بروتوكول منع الهلوسة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 22: بروتوكول اختيار الأداة

الغرض من بروتوكول اختيار الأداة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 23: بروتوكول ترقية الذاكرة

الغرض من بروتوكول ترقية الذاكرة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 24: بروتوكول ضغط التجربة

الغرض من بروتوكول ضغط التجربة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 25: بروتوكول قياس الجدة

الغرض من بروتوكول قياس الجدة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 26: بروتوكول محاكمة المصدر

الغرض من بروتوكول محاكمة المصدر هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 27: بروتوكول التخطيط السببي

الغرض من بروتوكول التخطيط السببي هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 28: بروتوكول تحويل المهارة إلى hyperskill

الغرض من بروتوكول تحويل المهارة إلى hyperskill هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 29: بروتوكول مقاومة benchmark gaming

الغرض من بروتوكول مقاومة benchmark gaming هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 30: بروتوكول بناء GraphWorld

الغرض من بروتوكول بناء GraphWorld هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 31: بروتوكول التفنيد العكسي

الغرض من بروتوكول التفنيد العكسي هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 32: بروتوكول أثر العالم

الغرض من بروتوكول أثر العالم هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 33: بروتوكول اختيار الفشل المفيد

الغرض من بروتوكول اختيار الفشل المفيد هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 34: بروتوكول الخريطة الوعائية

الغرض من بروتوكول الخريطة الوعائية هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 35: بروتوكول تحديث reward governance

الغرض من بروتوكول تحديث reward governance هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 36: بروتوكول اختبار few-shot transfer

الغرض من بروتوكول اختبار few-shot transfer هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 37: بروتوكول قياس تكلفة الخطأ

الغرض من بروتوكول قياس تكلفة الخطأ هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 38: بروتوكول استعمال مصادر متعارضة

الغرض من بروتوكول استعمال مصادر متعارضة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 39: بروتوكول المعرفة الزمنية

الغرض من بروتوكول المعرفة الزمنية هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 40: بروتوكول الإصدار والتجريب

الغرض من بروتوكول الإصدار والتجريب هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 41: بروتوكول منع الهلوسة

الغرض من بروتوكول منع الهلوسة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 42: بروتوكول اختيار الأداة

الغرض من بروتوكول اختيار الأداة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 43: بروتوكول ترقية الذاكرة

الغرض من بروتوكول ترقية الذاكرة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 44: بروتوكول ضغط التجربة

الغرض من بروتوكول ضغط التجربة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 45: بروتوكول قياس الجدة

الغرض من بروتوكول قياس الجدة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 46: بروتوكول محاكمة المصدر

الغرض من بروتوكول محاكمة المصدر هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 47: بروتوكول التخطيط السببي

الغرض من بروتوكول التخطيط السببي هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 48: بروتوكول تحويل المهارة إلى hyperskill

الغرض من بروتوكول تحويل المهارة إلى hyperskill هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 49: بروتوكول مقاومة benchmark gaming

الغرض من بروتوكول مقاومة benchmark gaming هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 50: بروتوكول بناء GraphWorld

الغرض من بروتوكول بناء GraphWorld هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 51: بروتوكول التفنيد العكسي

الغرض من بروتوكول التفنيد العكسي هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 52: بروتوكول أثر العالم

الغرض من بروتوكول أثر العالم هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 53: بروتوكول اختيار الفشل المفيد

الغرض من بروتوكول اختيار الفشل المفيد هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 54: بروتوكول الخريطة الوعائية

الغرض من بروتوكول الخريطة الوعائية هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 55: بروتوكول تحديث reward governance

الغرض من بروتوكول تحديث reward governance هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 56: بروتوكول اختبار few-shot transfer

الغرض من بروتوكول اختبار few-shot transfer هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 57: بروتوكول قياس تكلفة الخطأ

الغرض من بروتوكول قياس تكلفة الخطأ هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 58: بروتوكول استعمال مصادر متعارضة

الغرض من بروتوكول استعمال مصادر متعارضة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 59: بروتوكول المعرفة الزمنية

الغرض من بروتوكول المعرفة الزمنية هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 60: بروتوكول الإصدار والتجريب

الغرض من بروتوكول الإصدار والتجريب هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 61: بروتوكول منع الهلوسة

الغرض من بروتوكول منع الهلوسة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 62: بروتوكول اختيار الأداة

الغرض من بروتوكول اختيار الأداة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 63: بروتوكول ترقية الذاكرة

الغرض من بروتوكول ترقية الذاكرة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 64: بروتوكول ضغط التجربة

الغرض من بروتوكول ضغط التجربة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 65: بروتوكول قياس الجدة

الغرض من بروتوكول قياس الجدة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 66: بروتوكول محاكمة المصدر

الغرض من بروتوكول محاكمة المصدر هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 67: بروتوكول التخطيط السببي

الغرض من بروتوكول التخطيط السببي هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 68: بروتوكول تحويل المهارة إلى hyperskill

الغرض من بروتوكول تحويل المهارة إلى hyperskill هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 69: بروتوكول مقاومة benchmark gaming

الغرض من بروتوكول مقاومة benchmark gaming هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 70: بروتوكول بناء GraphWorld

الغرض من بروتوكول بناء GraphWorld هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 71: بروتوكول التفنيد العكسي

الغرض من بروتوكول التفنيد العكسي هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 72: بروتوكول أثر العالم

الغرض من بروتوكول أثر العالم هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 73: بروتوكول اختيار الفشل المفيد

الغرض من بروتوكول اختيار الفشل المفيد هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 74: بروتوكول الخريطة الوعائية

الغرض من بروتوكول الخريطة الوعائية هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 75: بروتوكول تحديث reward governance

الغرض من بروتوكول تحديث reward governance هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 76: بروتوكول اختبار few-shot transfer

الغرض من بروتوكول اختبار few-shot transfer هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 77: بروتوكول قياس تكلفة الخطأ

الغرض من بروتوكول قياس تكلفة الخطأ هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 78: بروتوكول استعمال مصادر متعارضة

الغرض من بروتوكول استعمال مصادر متعارضة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 79: بروتوكول المعرفة الزمنية

الغرض من بروتوكول المعرفة الزمنية هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 80: بروتوكول الإصدار والتجريب

الغرض من بروتوكول الإصدار والتجريب هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 81: بروتوكول منع الهلوسة

الغرض من بروتوكول منع الهلوسة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 82: بروتوكول اختيار الأداة

الغرض من بروتوكول اختيار الأداة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 83: بروتوكول ترقية الذاكرة

الغرض من بروتوكول ترقية الذاكرة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 84: بروتوكول ضغط التجربة

الغرض من بروتوكول ضغط التجربة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 85: بروتوكول قياس الجدة

الغرض من بروتوكول قياس الجدة هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 86: بروتوكول محاكمة المصدر

الغرض من بروتوكول محاكمة المصدر هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 87: بروتوكول التخطيط السببي

الغرض من بروتوكول التخطيط السببي هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

بطاقة 88: بروتوكول تحويل المهارة إلى hyperskill

الغرض من بروتوكول تحويل المهارة إلى hyperskill هو تحويل الشعار إلى خطوة تشغيلية. يبدأ البروتوكول بتحديد الدعوى أو الفعل، ثم استخراج الفروق الحاسمة، ثم تقدير MCD، ثم اختيار مسار: استرجاع، تفنيد، محاكاة، تنفيذ، أو توقف. لا يسمح البروتوكول بإضافة نتيجة إلى الذاكرة إلا بعد ظهور أثر أو دليل أو اعتراض تمت معالجته.

مرحلة	سؤال حاكم	مخرج
1. الإدخال	ما الدعوى أو المهمة؟	وصف مضبوط وحدود مجال
2. الفرق	ما الذي إذا تغير تغير القرار؟	قائمة فروق مرتبة
3. الحاكمية	ما المخاطر والتكلفة والجدة؟	MCD vector
4. الفعل	أي أداة أو تفكير أو مصدر؟	مسار قابل للتتبع
5. الهدم	كيف يفشل؟	اعتراض/مثال مضاد
6. الأثر	ماذا تغير في العالم أو الكود أو المعرفة؟	world-effect record
7. الذاكرة	ماذا يرقى؟ ماذا يحذف؟	memory patch

شرط النجاح: أن يخفض البروتوكول الأخطاء الحرجة أو ينقل مهارة إلى مهمة جديدة مع زيادة محدودة في التكلفة. شرط الفشل: أن يزيد التعقيد الشكلي دون أثر قابل للقياس. النسخة العلمية من MCAI-v2 لا تسمح لأي بروتوكول بالبقاء إذا لم ينجح في تجربة حذف/إضافة واضحة.

تقييم الجودة النهائي

المعيار	الدرجة	تعليل مختصر
عمق التأصيل	14/15	دمج causal، world models، GraphRAG، memory، agents، reasoning models، ARC، learning.
جودة المصادر	14/15	مصادر أساسية وحديثة؛ لا تغطي كل الأطروحات الجامعية بسبب حدود الوصول.
كثافة الصفحات	10/10	صفحات نصية وجداول وبطاقات ممتلئة.
وضوح البنية	10/10	أبواب، أدوات، نموذج، محكمة، برنامج.
قوة النقد	9/10	محكمة هدم واضحة؛ تحتاج تنفيذ تجريبي أعمق.
الأصالة والتركيب	14/15	MCAI-v2/M-VCGE وأدوات مساعف مركبة قابلة للاختبار.
قابلية الاختبار	9/10	مصفوفة benchmarks وablation؛ لا توجد نتائج تشغيلية حقيقية بعد.
جمال PDF	5/5	Amiri RTL A4 وجداول وصناديق.
عدم الحشو	5/5	كل بطاقة مرتبطة ببروتوكول أو أداة.
فائدة للقارئ المتقدم	5/5	ينتج برنامج بناء لا سردا عاما.
المجموع	95/100	نسخة بحثية قوية مع صدق حول حدود الدليل.

قائمة المصادر

[R1] OpenAI o1 System Card. OpenAI. 2024 /https://openai.com/index/openai-o1-system-card.
الأثر: نموذج مفكر، RL، تقييمات سلامة، حدود reasoning systems

[R2] DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. DeepSeek-AI. 2025 https://arxiv.org/abs/2501.12948.
الأثر: distillation، test-time reasoning، RL reasoning، حدود readability/language mixing

[R3] ReAct: Synergizing Reasoning and Acting in Language Models. Yao et al.. 2022/2023 https://arxiv.org/abs/2210.03629.
الأثر: التناوب بين reasoning/actions واستعمال البيئة لتقليل الهلوسة

[R4] Tree of Thoughts: Deliberate Problem Solving with Large Language Models. Yao et al.. 2023 https://arxiv.org/abs/2305.10601.
الأثر: بحث شجري على وحدات فكرية، backtracking، self-evaluation

[R5] Reflexion: Language Agents with Verbal Reinforcement Learning. Shinn et al.. 2023 https://arxiv.org/abs/2303.11366.
الأثر: تعلم لغوي من الفشل عبر ذاكرة انعكاسية دون تحديث الأوزان

[R6] From Local to Global: A Graph RAG Approach to Query-Focused Summarization. Edge et al. / Microsoft Research 2024. https://arxiv.org/abs/2404.16130.
الأثر: استخراج knowledge graph وcommunity summaries لتحسين RAG على corpora خاصة

[R7] Cognitive Architectures for Language Agents - CoALA. Sumers et al.. 2023/2024 https://arxiv.org/abs/2309.02427.
الأثر: ذاكرة معيارية، decision process، action space، هندسة وكلاء لغويين

[R8] Zhang et al.. 2024 .A Survey on the Memory Mechanism of Large Language Model based Agents https://arxiv.org/abs/ .2404.13501.
الأثر: أنماط ذاكرة الوكلاء، التقييم، التطبيقات، حدود الذاكرة الحالية

[R9] Huang et al.. 2024 .Understanding the planning of LLM agents: A survey .https://arxiv.org/abs/2402.02716.
الأثر: تخطيط الوكلاء: memory، reflection، external modules، selection، task decomposition

[R10] Cao et al.. 2025 .Large Language Models for Planning: A Comprehensive and Systematic Survey https://arxiv.org/abs/ .2505.19683.
الأثر: تصنيف تخطيط LLM بين search-based methods، finetuning، modules

[R11] Ding et al.. 2024/2025 .Understanding World or Predicting Future? A Comprehensive Survey of World Models .https://arxiv.org/abs/2411.14499.
الأثر: world models للفهم والتنبؤ، evaluation fragmentation، multimodal foundation models

[R12] Schölkopf et al.. 2021 .Towards Causal Representation Learning .https://arxiv.org/abs/2102.11107.
الأثر: الربط بين السببية والتعميم والتحويل، واكتشاف المتغيرات السببية من observations

[R13] François Chollet. 2019 .On the Measure of Intelligence .https://arxiv.org/abs/1911.01547.
الأثر: تعريف الذكاء ككفاءة اكتساب مهارة تحت priors وخبرة؛ ARC

[R14] Chollet et al.. 2025/2026 .ARC-AGI-2 and ARC Prize 2025 Technical Report. ARC Prize / https://arxiv.org/abs/ .2601.10904.
الأثر: قياس few-shot generalization؛ تقرير 2025: أعلى نتيجة خاصة 24% على ARC-AGI-2

[R15] Jimenez et al.. 2023/2024 .SWE-bench: Can Language Models Resolve Real-World GitHub Issues? https://arxiv.org/ .abs/2310.06770.
الأثر: اختبار هندسة برمجيات واقعي من GitHub؛ أساس SWE-bench Verified

[R16] Phuong et al.. 2024 .Evaluating Frontier Models for Dangerous Capabilities .https://arxiv.org/abs/2403.13793.
الأثر: منهج تقييم القدرات الخطرة: persuasion/deception/cyber/self-proliferation/self-reasoning

[R17] OpenAI. 2024 .OpenAI - Introducing SWE-bench Verified https://openai.com/index/introducing-swe-bench-verified/.
الأثر: نسخة بشرية التحقق من SWE-bench بـ 500 مسألة أكثر موثوقية

[R18] Microsoft. 2024-2026 .GraphRAG Documentation https://microsoft.github.io/graphrag/.
الأثر: Pipeline: graph extraction, community hierarchy, summaries, retrieval over graph structures

منهج استعمال المصادر

استعملت هذه المصادر لا بوصفها سلطة نهائية، بل بوصفها نقاط تثبيت للحد الأدنى العلمي: أعمال reasoning models تثبت اتجاه test-time compute وRL، وأعمال agents تثبت قيمة الذاكرة والأدوات والتخطيط، وأعمال GraphRAG/world models/causal representation تثبت أن المعرفة المسطحة لا تكفي للتعميم. من هذا الحد الأدنى يبدأ تركيب MCAI-v2 كفرضية فوق-نموذجية قابلة للهدم.

كل دعوى اختراعية في الكتاب - مثل VGB أو Novelty Entropy Meter أو Vascular Router - لم تنسب إلى المصدر بوصفها نتيجة جاهزة، بل صيغت كتركيب جديد يحتاج prototype. لذلك لا تخلط قائمة المصادر بين السند والاختراع: السند يثبت الفجوة أو الاتجاه، والاختراع يقترح جسرا هندسيا يجب أن يختبر.

القراءة الصارمة لهذه النسخة تقتضي إعادة تشغيل البروتوكولات على harness حقيقي، وحذف كل طبقة لا تحسن الأداء أو السلامة أو نقل المهارة. بهذا يبقى الكتاب برنامج بناء لا وثيقة ادعاء.