Moussaif Research-Warrior Architecture

— mcai-v2 agi new technology neoAGI neoAGI-v3

MCAI-v2 / HAKIM-MCD / Vascular Superposition / Moussaif Selective Assistance

زهير مساعف

Zouhair Moussaif

25 June 2026

كتاب بحثي عربي كثيف: نظريات، تقنيات، أدوات، محكمة هدم، وبرنامج قياس. كل benchmark score في هذا العمل هو محاكاة منهجية لا نتيجة رسمية.

صفحة العنوان والعقد الفكري

المؤلف الإطاري: زهير مساعف. السلسلة: Moussaif AGI Knowledge Architecture. المهمة: تحويل مشاريع MCAI/MIGAGI/HAKIM-MCD/Vascular Superposition/Selective Assistance إلى كتاب هندسي عربي كثيف، لا تجميعا ولا شرحا مدرسيا.

العقد: نكتب كما يبني مهندس أنظمة لا كما يعلّق مراقب. كل مفهوم يدخل الكتاب يجب أن يغير شيئا: تمييزا، مسارا، بروتوكولا، metric، أداة، اعتراضا، أو تجربة. الاسم الذي لا يؤدي وظيفة يزال.

اللغة عربية علمية مع إبقاء المصطلح الإنجليزي عند الحاجة: benchmark mutation, GraphRAG, verifier, test-time compute, tool-use. الغرض ليس زخرفة إنجليزية بل منع ضياع المعنى التقني.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تمهيد لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع صفحة العنوان والعقد الفكري نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة:** كثافة الصفحة ليست عدد كلمات فقط؛ إنها كثافة آليات وتمييزات واختبارات.

تنبيه منهجي: مراتب المعرفة داخل الكتاب

نستعمل ست مراتب: ثابت مدعوم بالمراجع، راجح علميا، خلافي أو فعال في سياق دون آخر، تركيب منطقي من مصادر، اختراع نموذجي لهذا الكتاب، وتأمل مولد للأسئلة. الخطأ الأكبر في مشاريع AGI هو خلط هذه المراتب.

ReAct وRAG وToolformer وTree of Thoughts وReflection وVoyager أدبيات لها شواهد منشورة. أما Vascular Compute Router وInventive Entropy Governance وMCD-Bench فهي تركيبات واختراعات مساعفية تحتاج قياسا.

هذا الفصل ليس اعتذارا عن الإبداع بل حماية له. الفكرة الجديدة تصبح محترمة حين تعلن أنها جديدة، وحين تسمح للخصم بتحديد الاختبار الذي قد يهدمها.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تمهيد لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع تنبيه منهجي: مراتب المعرفة داخل الكتاب نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع، يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

المرتبة	المعنى	طريقة التعامل
ثابت	تدعمه أدبيات أو معيار	يستخدم كقاعدة
راجح	قوي لكن مشروط	يذكر حد السياق
خلافي	مدارس أو نتائج متعارضة	يفتح مقارنة
تركيب	استنتاج من مصادر	يوسم كاستنتاج
اختراع	أداة/نظرية جديدة	يوضع اختبار
تأمل	مولد أسئلة	لا يستعمل كدليل

ملخص تنفيذي في صفحتين: المشكلة والحل

المشكلة: النماذج المفكرة الحالية أظهرت قوة في reasoning وtool use لكنها ما زالت تنهار عند السياق الطويل، تضارب الأدوات، ضعف التحقق، تلوث الذاكرة، وفشل نقل المهارة. الأدبيات لا تقول إن AGI انتهى؛ بل تقول إن طبقات جديدة من orchestration والقياس صارت ضرورية.

الحل المقترح: MCAI-v2 لا يجعل LLM مركز الكون، بل يجعله عضوا داخل نظام حاكمي. يبدأ النظام من فرق معرفي Δ لا من prompt، يحسب MCD، يختار مسارات تفكير وأدوات، يستدعي GraphRAG مع أدلة، يمرر الناتج على Perturbation Court، ينفذ World-Effect Loop، ثم يكتب Self-Reconstruction Memory.

القفزة ليست أكبر context فقط. القفزة هي أن يصبح للنظام جهاز انتقاء: يعرف أي سؤال يحتاج بحثا، أي مسار يحتاج محاكاة، أي أداة خطرة، أي ذاكرة يجب رفضها، وأي benchmark يجب تحويره حتى لا يخدع النظام بنجاح محلي.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ملخص تنفيذي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع ملخص تنفيذي في صفحتين: المشكلة والحل نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **صيغة مختصرة:** Δ → MCD → Vascular Routing → Tool/GraphRAG → Court → Agency Selection → World Effect → Memory Rewrite.

خريطة مفاهيمية عامة

المسار المفاهيمي يبدأ من الفرق: ما الذي تغير، تناقض، غاب، أو صار غير قابل للحسم؟ الفرق يدخل MCD بوصفه مقياسا مركبا: كثافة الفروق، تفاعل القيود، العمق الزمني، عبء الوكالة، ضغط الدليل، واضطراب السياق.

إذا كان MCD منخفضا، يكفي جواب قصير مع مصدر. إذا كان عاليا، يفتح النظام أوعية: وعاء مصادر، وعاء سببية، وعاء أدوات، وعاء محكمة، وعاء تطبيق. هذه هي الوعائية: عدم اختزال المسألة في قناة واحدة.

الحاكمية هي اختيار من يملك الحق في قيادة المسار: الدليل، المثال، الأداة، الذاكرة، المحاكاة، أو الاعتراض. لا ذكاء عام بلا حاكمية؛ لأن القدرة بلا انتقاء تتحول إلى ضجيج.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تمهيد لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع خريطة مفاهيمية عامة نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

خريطة نصية: Causal Map → Mechanism Extraction → Concept Definition → Evidence Rank → Source Ledger → Research Question → Benchmark → Protocol.

فهرس تفصيلي 1

الفهرس هنا ليس زخرفة؛ إنه خريطة حمل معرفي. كل باب مصمم كطبقة من سبع: تعريف، نسب تاريخي، مدارس، آليات، فشل، تركيب، برنامج اختبار.

القراءة الصحيحة للكتاب تمر بثلاث حركات: مسح بنيوي، استخراج أدوات، ثم تحويل الأدوات إلى prototype. القارئ الباحث لا يسأل هل الاسم جميل؛ يسأل ماذا يقيس، ماذا يمنع، ومتى يفشل.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تمهيد لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع فهرس تفصيلي 1 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

الموضع	العنوان	وظيفة الباب
الباب 1	العقد المنهجي ومراتب الدليل	تثبيت الحدود بين العلم، التركيب، الفرضية، والمحاكاة.
الباب 2	تشريح البنية الحالية للنماذج المفكرة	ما تعمل به LLM agents الآن: reasoning, tools, memory, RAG, verifiers.
الباب 3	أين تفشل البنيات الحالية	فشل السببية، هشاشة الأدوات، الذاكرة، التخطيط، والبنچمارك.
الباب 4	MCAI-v2 كمعمار حاكمي	فرق، MCD، حاكمية، محكمة، أثر، ذاكرة.

فهرس تفصيلي 2

الفهرس هنا ليس زخرفة؛ إنه خريطة حمل معرفي. كل باب مصمم كطبقة من سبع: تعريف، نسب تاريخي، مدارس، آليات، فشل، تركيب، برنامج اختبار.

القراءة الصحيحة للكتاب تمر بثلاث حركات: مسح بنيوي، استخراج أدوات، ثم تحويل الأدوات إلى prototype. القارئ الباحث لا يسأل هل الاسم جميل؛ يسأل ماذا يقيس، ماذا يمنع، ومتى يفشل.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تمهيد لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع فهرس تفصيلي 2 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

الموضع	العنوان	وظيفة الباب
الباب 5	ثلاث نظريات جديدة	توليد نظري مضبوط لا ادعاء إثبات.
الباب 6	ثلاث تقنيات تنفيذية	من النظرية إلى OS وGraphRAG ومصنع معايير.
الباب 7	خمسة عشر أداة AGI تقنية	أدوات قابلة للبروتوتايب والقياس.
الباب 8	فلسفات وأدوات التدريب	كيف ندرّب النموذج على الحاكمية والفشل والسببية.

فهرس تفصيلي 3

الفهرس هنا ليس زخرفة؛ إنه خريطة حمل معرفي. كل باب مصمم كطبقة من سبع: تعريف، نسب تاريخي، مدارس، آليات، فشل، تركيب، برنامج اختبار.

القراءة الصحيحة للكتاب تمر بثلاث حركات: مسح بنيوي، استخراج أدوات، ثم تحويل الأدوات إلى prototype. القارئ الباحث لا يسأل هل الاسم جميل؛ يسأل ماذا يقيس، ماذا يمنع، ومتى يفشل.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تمهيد لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع فهرس تفصيلي 3 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

الموضع	العنوان	وظيفة الباب
الباب 9	محاكاة benchmark 13	scores نظرية مميزة بأنها غير رسمية وغير منفذة.
الباب 10	محكمة الهدم والبرنامج البحثي	اعتراضات، اختبارات، فرضيات، مشاريع، وتحسينات.

سجل المصادر المركزي: لماذا هذه العائلات؟

اختيار المصادر لم يكن لتكثير الهوامش. العائلة الأولى هي أدبيات reasoning/action والوكالة: ReAct, Reflexion, ToT, Voyager, Toolformer, MRKL. العائلة الثانية هي الذاكرة والمعرفة: GraphRAG/ontologies, Generative Agents, RAG. العائلة الثالثة هي benchmarks التي تكشف حدود المجال.

العائلة الرابعة هي أدوات الصناعة الحديثة: Agents SDK, MCP, tool calling, guardrails, tracing. العائلة الخامسة هي أعمال مساعف السابقة: المعرفة كـMCD، hyperskills، evidence ranks، engine، محكمة الهدم، والانتقاء الاختراعي.

أثر المصادر في الكتاب ليس تلخيصا. كل مصدر يتحول إلى آلية: ReAct يصبح Reflexion، interleaving تصبح ذاكرة فشل، ToT يصبح بحث مسارات، Voyager يصبح SWE-bench، skill library يصبح واقع هندسة برمجية، GAIA يصبح مساعدا عاما بأدوات.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب مصادر لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع سجل المصادر المركزي: لماذا هذه العائلات؟ نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بـtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **منهج:** المصدر الذي لا يتحول إلى آلية أو حد أو اختبار يبقى في الببليوغرافيا ولا يدخل المعمار.

جدول مصادر عامل 1

هذا الجدول يربط كل مصدر بوظيفته داخل البناء. لا يهم أن يكون المصدر مشهورا إن لم يغير تصميم النظام أو معيار اختباره.

المصدر المركزي يعطينا آلية. المصدر المساعد يعطينا حدا أو مقارنة. المصدر الإشكالي يعطينا اعتراضا. المصدر الحديث يعطينا حالة المجال، لكنه لا يصبح حقيقة مطلقة لمجرد حداثته.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب مصادر لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع جدول مصادر عامل 1 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

رمز	المصدر	المؤلف/الجهة	السنة	وظيفته
S01	ReAct: Synergizing Reasoning and Acting in Language Models	Yao et al.	2022	reasoning-action interleaving
S02	Reflexion: Language Agents with Verbal Reinforcement Learning	Shinn et al.	2023	verbal feedback, episodic memory
S03	Tree of Thoughts: Deliberate Problem Solving with LLMs	Yao et al.	2023	search over reasoning paths
S04	Voyager: An Open-Ended Embodied Agent with LLMs	Wang et al.	2023	skill library and automatic curriculum
S05	Toolformer: Language Models Can Teach Themselves to Use Tools	Schick et al.	2023	tool-use decisions from self-supervision
S06	MRKL Systems	Karpas et al.	2022	neuro-symbolic modular routing
S07	Retrieval-Augmented Generation for Knowledge-Intensive NLP	Lewis et al.	2020	parametric + non-parametric memory
S08	Generative Agents: Interactive Simulacra of Human Behavior	Park et al.	2023	observation, memory, reflection, planning

جدول مصادر عامل 2

هذا الجدول يربط كل مصدر بوظيفته داخل البناء. لا يهم أن يكون المصدر مشهورا إن لم يغير تصميم النظام أو معيار اختباره.

المصدر المركزي يعطينا آلية. المصدر المساعد يعطينا حدا أو مقارنة. المصدر الإشكالي يعطينا اعتراضا. المصدر الحديث يعطينا حالة المجال، لكنه لا يصبح حقيقة مطلقة لمجرد حداثته.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب مصادر لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع جدول مصادر عامل 2 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

رمز	المصدر	المؤلف/الجهة	السنة	وظيفته
S09	AgentBench: Evaluating LLMs as Agents	Liu et al.	2023/2024	multi-environment agent evaluation
S10	WebArena: A Realistic Web Environment for Autonomous Agents	Zhou et al.	2023	realistic web tasks and tool-use
S11	SWE-bench: Can LMs Resolve Real-World GitHub Issues?	Jimenez et al.	2023/2024	codebase issue resolution benchmark
S12	GAIA: a benchmark for General AI Assistants	Mialon et al.	2023	tool use, browsing, multimodal assistant tasks
S13	MMLU-Pro	Wang et al.	2024	harder reasoning-focused multitask benchmark
S14	GPQA	Rein et al.	2023	graduate-level Google-proof science questions
S15	FrontierMath	Glazer et al.	2024	expert mathematics frontier benchmark
S16	Humanity's Last Exam	Center for AI Safety / Scale AI	2025	frontier closed-ended expert knowledge

جدول مصادر عامل 3

هذا الجدول يربط كل مصدر بوظيفته داخل البناء. لا يهم أن يكون المصدر مشهورا إن لم يغير تصميم النظام أو معيار اختباره.

المصدر المركزي يعطينا آلية. المصدر المساعد يعطينا حدا أو مقارنة. المصدر الإشكالي يعطينا اعتراضا. المصدر الحديث يعطينا حالة المجال، لكنه لا يصبح حقيقة مطلقة لمجرد حداثته.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب مصادر لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع جدول مصادر عامل 3 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

رمز	المصدر	المؤلف/الجهة	السنة	وظيفته
S17	ARC-AGI-2	ARC Prize	2025/2026	abstract reasoning and efficiency
S18	OSWorld	OSWorld team	2024/2025	open-ended computer-use tasks
S19	OpenAI Agents SDK / Responses API tools	OpenAI	2025/2026	tools, guardrails, tracing, orchestration
S20	Model Context Protocol	Anthropic / MCP community	2024/2026	open standard for connecting AI apps to tools
S21	Breaking the Protocol: MCP Security Analysis	Maloyan & Namiot	2026	tool-integrated agent security vulnerabilities
S22	Benchmark Test-Time Scaling of General LLM Agents	.Li et al	2026	context ceiling and verification gap
S23	Moussaif Cognitive Knowledge Base Engine 430p	Moussaif prior work	2026	evidence ranks, anti-summary, knowledge engine
S24	Moussaif Hyperskill Loop Theory 560p	Moussaif prior work	2026	loop-skill-hyperskill governance

الباب الأول / طبقة المنهج 1: Difference Δ

المنهج يبدأ من Difference Δ: الفرق النشط هو بداية التفكير لا السؤال السطحي. الفرق قد يكون نقص دليل، تناقض نتائج، فجوة سببية، خطر أداة، أو انتقال مجال. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل كخلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا؛ نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 1: Difference Δ نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج 2: MCD

المنهج يبدأ من MCD: مقياس تعقيد مركب يرفع كلفة التفكير حين تتفاعل القيود والوكالة والسببية والزمان. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل تُحلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا، نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 2: MCD نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج 3: Vascular Routing

المنهج يبدأ من Vascular Routing: توجيه معرفي متعدد الأوعية: مصادر، سببية، أدوات، ذاكرة، محكمة، أثر. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل تُحلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا، نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 3: Vascular Routing نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج 4: Epistemic Governance

المنهج يبدأ من Epistemic Governance: حاكمية تحدد متى تقود الأدلة ومتى تقود المحاكاة ومتى يجب الإيقاف. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل تكلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا، نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 4: Epistemic Governance نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج 5: Perturbation Court

المنهج يبدأ من Perturbation Court: محكمة تدخل اعتراضات قوية قبل اعتماد نتيجة أو إدخالها في الذاكرة. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل تحلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا؛ نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 5: Perturbation Court نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج 6: Self-Reconstruction Memory

المنهج يبدأ من Self-Reconstruction Memory: ذاكرة لا تخزن جوابا فقط بل سبب نجاحه وفشله وحدوده. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل تكلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا؛ نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 6: Self-Reconstruction Memory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بـtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج 7: World-Effect Loop

المنهج يبدأ من World-Effect Loop: ربط التفكير بتأثير قابل للرصد: كود يعمل، تجربة، قرار، قياس. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل تكلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا، نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 7: World-Effect Loop نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج 8: Benchmark Mutation

المنهج يبدأ من Benchmark Mutation: تحويل الاختبار إلى عائلة مقاومة للحفظ والتلوث والتكيّف السطحي. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل تُحلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا؛ نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 8: Benchmark Mutation نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج 9: Difference Δ

المنهج يبدأ من Difference Δ: الفرق النشط هو بداية التفكير لا السؤال السطحي. الفرق قد يكون نقص دليل، تناقض نتائج، فجوة سببية، خطر أداة، أو انتقال مجال. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل كخلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا؛ نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 9: Difference Δ نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج MCD :10

المنهج يبدأ من MCD: مقياس تعقيد مركب يرفع كلفة التفكير حين تتفاعل القيود والوكالة والسببية والزمان. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل تُحلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا؛ نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج MCD :10 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج 11: Vascular Routing

المنهج يبدأ من Vascular Routing: توجيه معرفي متعدد الأوعية: مصادر، سببية، أدوات، ذاكرة، محكمة، أثر. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل تُحلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا؛ نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 11: Vascular Routing نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج 13: Perturbation Court

المنهج يبدأ من Perturbation Court: محكمة تدخل اعتراضات قوية قبل اعتماد نتيجة أو إدخالها في الذاكرة. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل تُحلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا؛ نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 13: Perturbation Court نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج 14: Self-Reconstruction Memory

المنهج يبدأ من Self-Reconstruction Memory: ذاكرة لا تخزن جوابا فقط بل سبب نجاحه وفشله وحدوده. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل تكلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا؛ نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 14: Self-Reconstruction Memory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج 15: World-Effect Loop

المنهج يبدأ من World-Effect Loop: ربط التفكير بتأثير قابل للرصد: كود يعمل، تجربة، قرار، قياس. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل كخلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا، نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 15: World-Effect Loop نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الأول / طبقة المنهج 16: Benchmark Mutation

المنهج يبدأ من Benchmark Mutation: تحويل الاختبار إلى عائلة مقاومة للحفظ والتلوث والتكيّف السطحي. في MCAI-v2 لا تُقرأ المسألة كطلب جواب بل تُحلل في نظام معرفة يحتاج موضعا، رتبة، آلية، وحد اختبار.

المعلوم الثابت في المجال: النماذج اللغوية تستفيد من التدرج في التفكير، من الأدوات، ومن الذاكرة الخارجية في مهام كثيرة. الراجح: التحقق والبحث المساري يزيدان الاستقرار في مسائل معينة. الخلافي: هل تكفي هذه الطبقات لإنتاج ذكاء عام أم أنها أقنعة حول نموذج إحصائي قوي؟

التركيب المساعفي: لا نطلب من LLM أن يكون عقلا كاملا؛ نطلب منه أن يعمل داخل دستور حاكمي. إذا تعارضت السلاسة مع الدليل ينتصر الدليل؛ إذا تعارضت السرعة مع الخطر ينتصر الإيقاف؛ إذا تعارضت novelty مع الآلية تنتصر المحكمة.

الاعتراض القوي: قد تكون هذه اللغة طبقات orchestration مع أسماء جديدة. الجواب: الاسم لا يهم إلا بقدر ما ينتج ablation test. إن لم يرفع Vascular Routing الدقة أو الاستقرار أو نقل المهارة عند الاختبار، فهو زخرفة ويجب إسقاطه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب العقد المنهجي ومراتب الدليل لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الأول / طبقة المنهج 16: Benchmark Mutation نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **اختبار:** قارن نظاما بلا محكمة ولا MCD بنظام MCAI-v2 على GAIA/WebArena/SWE-bench مع نفس النموذج والميزانية.

الباب الثاني / تشريح الحدود الحالية 2: Reflexion

Reflexion: يستبدل تحديث الأوزان بذاكرة لفظية للفشل؛ قيمته في التعلم عبر التجربة دون تدريب جديد. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Reflexion من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما؛ نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا، والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 2: Reflexion نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S02 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 3: Tree of Thoughts

Tree of Thoughts: يفتح بحثا فوق مسارات تفكير متعددة، قيمته في المسائل التي تحتاج backtracking واختيارا عالميا. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Tree of Thoughts من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما، نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا، والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 3: Tree of Thoughts نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع، يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S03 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 4: Voyager

Voyager: يحوّل التجربة إلى skill library؛ أهميته أن المهارة تصبح قابلة للتركيب والاستدعاء. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Voyager من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما؛ نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا؛ والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 4: Voyager نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S04 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 5: Toolformer/MRKL

Toolformer/MRKL: ينقل النموذج من متكلم إلى router فوق أدوات وخبراء؛ الخطر هو الثقة الزائدة في tool outputs. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Toolformer/MRKL من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما؛ نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا؛ والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 5: Toolformer/MRKL نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S05/S06 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 6: RAG/GraphRAG

RAG/GraphRAG: يفتح ذاكرة غير معلمية؛ حدوده في retrieval drift، تضارب المصادر، وضعف causal compression. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول RAG/GraphRAG من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما؛ نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا، والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 6: RAG/GraphRAG نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع، يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S07 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 8: Test-time scaling

Test-time scaling: يزيد البحث أثناء الاستدلال، حدوده تظهر عند سقف السياق وفجوة التحقق. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Test-time scaling من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما، نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا، والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا، يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 8: Test-time scaling نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع، يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم، الثانية تضيف schema، الثالثة تربطه بtrace، الرابعة تقيسه بأبلشن، الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة، وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S22 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 9: ReAct

ReAct: يمزج reasoning/action في مسار واحد؛ قوته أنه يجعل التفكير يتغذى من ملاحظة خارجية بدل الانغلاق داخل النص. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول ReAct من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما؛ نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا، والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 9: ReAct نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S01 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 10: Reflexion

Reflexion: يستبدل تحديث الأوزان بذاكرة لفظية للفشل؛ قيمته في التعلم عبر التجربة دون تدريب جديد. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Reflexion من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما؛ نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا، والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 10: Reflexion نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S02 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 11: Tree of Thoughts

Tree of Thoughts: يفتح بحثا فوق مسارات تفكير متعددة، قيمته في المسائل التي تحتاج backtracking واختيارا عالميا. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Tree of Thoughts من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما، نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا، والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 11: Tree of Thoughts نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع، يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S03 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 12: Voyager

Voyager: يحوّل التجربة إلى skill library؛ أهميته أن المهارة تصبح قابلة للتركيب والاستدعاء. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Voyager من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما؛ نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا؛ والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 12: Voyager نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S04 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 14: RAG/GraphRAG

RAG/GraphRAG: يفتح ذاكرة غير معلمية؛ حدوده في retrieval drift، تضارب المصادر، وضعف causal compression. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول RAG/GraphRAG من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما؛ نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا، والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 14: RAG/GraphRAG نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S07 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 15: Agents SDK/MCP

Agents SDK/MCP: يحول الوكالة إلى بروتوكولات وأدوات وtracing، حدوده في الأمن والصلاحيات والحقن. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Agents SDK/MCP من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما، نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا، والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا، يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 15: Agents SDK/MCP نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع، يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم، الثانية تضيف schema، الثالثة تربطه بtrace، الرابعة تقيسه بأبلشن، الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة، وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S19/S20/S21 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 16: Test-time scaling

Test-time scaling: يزيد البحث أثناء الاستدلال، حدوده تظهر عند سقف السياق وفجوة التحقق. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Test-time scaling من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما، نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا، والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 16: Test-time scaling نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع، يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم، الثانية تضيف schema، الثالثة تربطه بtrace، الرابعة تقيسه بأبلشن، الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة، وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S22 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 17: ReAct

ReAct: يمزج reasoning/action في مسار واحد؛ قوته أنه يجعل التفكير يتغذى من ملاحظة خارجية بدل الانغلاق داخل النص. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول ReAct من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما؛ نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا؛ والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 17: ReAct نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S01 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 19: Tree of Thoughts

Tree of Thoughts: يفتح بحثا فوق مسارات تفكير متعددة، قيمته في المسائل التي تحتاج backtracking واختيارا عالميا. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Tree of Thoughts من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما؛ نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا، والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 19: Tree of Thoughts نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S03 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 20: Voyager

Voyager: يحوّل التجربة إلى skill library؛ أهميته أن المهارة تصبح قابلة للتركيب والاستدعاء. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Voyager من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما؛ نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا؛ والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 20: Voyager نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S04 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 21: Toolformer/MRKL

Toolformer/MRKL: ينقل النموذج من متكلم إلى router فوق أدوات وخبراء؛ الخطر هو الثقة الزائدة في tool outputs. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Toolformer/MRKL من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما؛ نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا؛ والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 21: Toolformer/MRKL نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S05/S06 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 22: RAG/GraphRAG

RAG/GraphRAG: يفتح ذاكرة غير معلمية؛ حدوده في retrieval drift، تضارب المصادر، وضعف causal compression. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول RAG/GraphRAG من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما؛ نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا؛ والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 22: RAG/GraphRAG نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S07 في سجل المصادر.

الباب الثاني / تشريح الحدود الحالية 23: Agents SDK/MCP

Agents SDK/MCP: يحول الوكالة إلى بروتوكولات وأدوات وtracing، حدوده في الأمن والصلاحيات والحقن. هذا ليس AGI بذاته، لكنه قطعة في معمار أوسع. الخطأ أن نعبد القطعة: prompt engineering وحده لا يكفي، RAG وحده لا يكفي، tool use وحده لا يكفي، ولا ذاكرة غير محكومة تكفي.

الدرس المشترك من الأدبيات: النموذج يحتاج بيئة، أداة، ذاكرة، verifier، وسياسة توقف. كلما طال الأفق زادت أخطاء الحالة، وتراكمت التبعات الصغيرة حتى يصبح النجاح الأولي وهما.

في MCAI-v2 يتحول Agents SDK/MCP من تقنية منفصلة إلى وعاء داخل vascular system. لا نضيفه دائما، نضيفه عندما يطلبه MCD. المسألة السهلة لا تستحق بحثا شجريا، والمسألة الخطرة لا يسمح لها بأداة بلا Firewall.

الفجوة الخفية: معظم papers تقيس تحسنا داخل benchmark محدد. AGI يحتاج استقرارا عبر عائلات متحورة من الاختبارات، عبر أدوات مختلفة، ومع ذاكرة قابلة للتلوث. لهذا ينتقل الكتاب من benchmark score إلى benchmark survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب تشريح البنية الحالية للنماذج المفكرة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا، يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثاني / تشريح الحدود الحالية 23: Agents SDK/MCP نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع، يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم، الثانية تضيف schema، الثالثة تربطه بtrace، الرابعة تقيسه بأبلشن، الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة، وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مصدر محوري:** يرتبط هذا التحليل بـ S19/S20/S21 في سجل المصادر.

الباب الثالث / أطلس الفشل 1: فشل الحالة State Failure

فشل الحالة State Failure: النظام لا يعرف ما الحالة الحقيقية للعالم أو للمهمة بعد عدة خطوات. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 1: فشل الحالة State Failure نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 2: فشل التخطيط Long-horizon Planning

فشل التخطيط Long-horizon Planning: ينتج خطة لغوية لكنها لا تصمد أمام تغيّر الملاحظات. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 2: فشل التخطيط Long-horizon Planning نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 3: فشل الأداة Tool Misuse

فشل الأداة Tool Misuse: يستدعي أداة صحيحة في وقت خاطئ أو بإدخال خطير. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 3: فشل الأداة Tool Misuse تفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 4: فشل التحقق Verification Gap

فشل التحقق Verification Gap: ينتج عدة مسارات ولا يملك verifier كافيا لاختيار الصحيح. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 4: فشل التحقق Verification Gap نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 6: فشل السببية Causal Blindness

فشل السببية Causal Blindness: يخلط الترابط بالسبب ويستخرج نتيجة بلا آلية. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 6: فشل السببية Causal Blindness نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 7: فشل النقل Transfer Failure

فشل النقل Transfer Failure: ينجح في benchmark ويضعف في توزيع جديد. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 7: فشل النقل Transfer Failure نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 8: فشل الحوكمة Permission Drift

فشل الحوكمة Permission Drift: يتوسع في الصلاحيات لأن الأداة متاحة لا لأنها مأذونة. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 8: فشل الحوكمة Permission Drift نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 9: فشل الحالة State Failure

فشل الحالة State Failure: النظام لا يعرف ما الحالة الحقيقية للعالم أو للمهمة بعد عدة خطوات. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 9: فشل الحالة State Failure نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 10: فشل التخطيط Long-horizon Planning

فشل التخطيط Long-horizon Planning: ينتج خطة لغوية لكنها لا تصمد أمام تغيّر الملاحظات. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 10: فشل التخطيط Long-horizon Planning نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 12: فشل التحقق Verification Gap

فشل التحقق Verification Gap: ينتج عدة مسارات ولا يملك verifier كافيا لاختيار الصحيح. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 12: فشل التحقق Verification Gap نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 13: فشل الذاكرة Memory Pollution

فشل الذاكرة Memory Pollution: يحفظ تأملات أو أخطاء كأنها خبرة موثوقة. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 13: فشل الذاكرة Memory Pollution نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 14: فشل السببية Causal Blindness

فشل السببية Causal Blindness: يخلط الترابط بالسبب ويستخرج نتيجة بلا آلية. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 14: فشل السببية Causal Blindness نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 15: فشل النقل Transfer Failure

فشل النقل Transfer Failure: ينجح في benchmark ويضعف في توزيع جديد. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 15: فشل النقل Transfer Failure نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 17: فشل الحالة State Failure

فشل الحالة State Failure: النظام لا يعرف ما الحالة الحقيقية للعالم أو للمهمة بعد عدة خطوات. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 17: فشل الحالة State Failure نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 18: فشل التخطيط Long-horizon Planning

فشل التخطيط Long-horizon Planning: ينتج خطة لغوية لكنها لا تصمد أمام تغيّر الملاحظات. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 18: فشل التخطيط Long-horizon Planning نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 19: فشل الأداة Tool Misuse

فشل الأداة Tool Misuse: يستدعي أداة صحيحة في وقت خاطئ أو بإدخال خطير. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 19: فشل الأداة Tool Misuse نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 20: فشل التحقق Verification Gap

فشل التحقق Verification Gap: ينتج عدة مسارات ولا يملك verifier كافيا لاختيار الصحيح. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 20: فشل التحقق Verification Gap نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الثالث / أطلس الفشل 21: فشل الذاكرة Memory Pollution

فشل الذاكرة Memory Pollution: يحفظ تأملات أو أخطاء كأنها خبرة موثوقة. هذا الفشل ليس خطأ عارضا بل نمط بنيوي. كل نظام agentic طويل الأفق يجب أن يملك طريقة لتسمية هذا الفشل قبل أن يدعي إصلاحه.

الحل المساعفي ليس زيادة التعليمات بل بناء أثر تشخيصي: perception → state schema → plan → tool call → observation → verifier → memory write → stop rule. إذا لم نعرف أين وقع الخلل فلن نطور النظام بل سنبدل prompt.

التقنية الجديدة المطلوبة هنا هي Failure Atlas مرتبط بالذاكرة. كل فشل يصبح object: trigger, context, wrong assumption, missing evidence, failed verifier, repair, retirement condition. حين يتكرر الفشل يتحول إلى hyperskill جديدة.

اعتراض: قد يزيد هذا العبء الكلفة والبطء. الجواب: نعم، ولهذا نحتاج MCD Router. لا تطبق المحكمة الكاملة على كل شيء؛ تطبق حيث عبء الخطأ أو قيمة المعرفة يبرران الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب أين تفشل البنيات الحالية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثالث / أطلس الفشل 21: فشل الذاكرة Memory Pollution نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة هندسية:** كل failure بلا سجل يصبح قدرا؛ وكل سجل بلا repair يصبح أرشيفا ميتا.

الباب الرابع / MCAI-v2 module 2: Meta-Complexity Mapper

Meta-Complexity Mapper: يحسب MCD ويحدد عمق التفاعل بين القيود. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 2: Meta-Complexity Mapper نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 3: Inner Time Engine

Inner Time Engine: يرتب المسألة زمنيا: ما قبلها، أثناءها، آثارها، ونقاط الرجوع. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 3: Inner Time Engine نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 4: Vascular Compute Router

Vascular Compute Router: يوزع compute على الأوعية ذات العائد الأكبر. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 4: Vascular Compute Router نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 5: Global Workspace

Global Workspace: يجمع نتائج المسارات في مساحة منافسة منظمة. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 5: Global Workspace نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 6: Agency Selector

Agency Selector: يختار الفعل: جواب، بحث، أداة، توقف، تصعيد، أو تجربة. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 6: Agency Selector نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 8: World-Effect Loop

World-Effect Loop: يطلب أثرا قابلا للقياس لا نصا فقط. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 8: World-Effect Loop نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 9: Self-Reconstruction Memory

Self-Reconstruction Memory: يعيد كتابة هوية النظام العملية من أسباب النجاح والفشل. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 9: Self-Reconstruction Memory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 10: Difference Extractor

Difference Extractor: يستخرج الفروق النشطة من الطلب والمصادر والسياق. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 10: Difference Extractor نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 11: Meta-Complexity Mapper

Meta-Complexity Mapper: يحسب MCD ويحدد عمق التفاعل بين القيود. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 11: Meta-Complexity Mapper نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 13: Vascular Compute Router

Vascular Compute Router: يوزع compute على الأوعية ذات العائد الأكبر. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 13: Vascular Compute Router نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 14: Global Workspace

Global Workspace: يجمع نتائج المسارات في مساحة منافسة منظمة. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 14: Global Workspace نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 15: Agency Selector

Agency Selector: يختار الفعل: جواب، بحث، أداة، توقف، تصعيد، أو تجربة. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 15: Agency Selector نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 16: Perturbation Court

Perturbation Court: يهاجم النتيجة قبل إخراجها. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 16: Perturbation Court نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 17: World-Effect Loop

World-Effect Loop: يطلب أثرا قابلا للقياس لا نصا فقط. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 17: World-Effect Loop نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 19: Difference Extractor

Difference Extractor: يستخرج الفروق النشطة من الطلب والمصادر والسياق. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 19: Difference Extractor نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 20: Meta-Complexity Mapper

Meta-Complexity Mapper: يحسب MCD ويحدد عمق التفاعل بين القيود. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 20: Meta-Complexity Mapper نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 21: Inner Time Engine

Inner Time Engine: يرتب المسألة زمنيا: ما قبلها، أثناءها، آثارها، ونقاط الرجوع. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 21: Inner Time Engine نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 22: Vascular Compute Router

Vascular Compute Router: يوزع compute على الأوعية ذات العائد الأكبر. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 22: Vascular Compute Router نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 24: Agency Selector

Agency Selector: يختار الفعل: جواب، بحث، أداة، توقف، تصعيد، أو تجربة. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 24: Agency Selector نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 25: Perturbation Court

Perturbation Court: يهاجم النتيجة قبل إخراجها. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 25: Perturbation Court نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 26: World-Effect Loop

World-Effect Loop: يطلب أثرا قابلا للقياس لا نصا فقط. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 26: World-Effect Loop نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 27: Self-Reconstruction Memory

Self-Reconstruction Memory: يعيد كتابة هوية النظام العملية من أسباب النجاح والفشل. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 27: Self-Reconstruction Memory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 28: Difference Extractor

Difference Extractor: يستخرج الفروق النشطة من الطلب والمصادر والسياق. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 28: Difference Extractor نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الرابع / MCAI-v2 module 30: Inner Time Engine

Inner Time Engine: يرتب المسألة زمنيا: ما قبلها، أثناءها، آثارها، ونقاط الرجوع. داخل MCAI-v2 هذه ليست دالة مساعدة بل موضع حاكمي. وظيفتها أن تمنع النموذج من الانتقال من اللغة إلى القرار بلا وسيط معرفي.

مدخل الوحدة: نص الطلب، سياق المستخدم، مصادر مسترجعة، قيود السلامة، وسجل ذاكرة. مخرجها: قرار مبرر، trace، evidence rank، وشرط إيقاف. كل وحدة يجب أن تكون قابلة للقياس بعزلها أو تعطيلها.

آلية التشغيل: إذا زاد MCD، تُفتح مسارات متعددة وتدخل الأدوات. إذا انخفض MCD، تُختصر الدورة. إذا ظهرت صلاحية خطرة، تتقدم Tool-Risk Firewall. إذا تعارضت المصادر، يتقدم Causal Map Compiler وPerturbation Court.

حدود المعرفة: لا يوجد دليل أن هذا المعمار يحقق AGI. الدليل الممكن هو تفوق مستقر عبر عائلات benchmark mutated مع نفس نموذج الأساس. إذن MCAI-v2 فرضية هندسية قوية، لا حقيقة مثبتة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب MCAI-v2 كمعمار حاكمي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الرابع / MCAI-v2 module 30: Inner Time Engine نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر قياس:** تحسن success stability، انخفاض hallucination، ارتفاع recovery after failure، انخفاض unsafe tool calls.

الباب الخامس / النظرية 1.1: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory

التعريف: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory. الأطروحة: الذكاء العام ليس اتساعا في الإجابة بل قدرة على تحويل الفروق Δ إلى مسارات سببية قابلة للاختبار، ثم حوكمتها عبر MCD قبل الفعل. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بتقليل أخطاء السببية، زيادة نجاح tool-use، وتحسن نقل المهارة بين مجالات لا تشترك في القاموس. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 1.1: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 1.2: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory

التعريف: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory. الأطروحة: الذكاء العام ليس اتساعا في الإجابة بل قدرة على تحويل الفروق Δ إلى مسارات سببية قابلة للاختبار، ثم حوكمتها عبر MCD قبل الفعل. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بتقليل أخطاء السببية، زيادة نجاح tool-use، وتحسن نقل المهارة بين مجالات لا تشترك في القاموس. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 1.2: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 1.3: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory

التعريف: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory. الأطروحة: الذكاء العام ليس اتساعا في الإجابة بل قدرة على تحويل الفروق Δ إلى مسارات سببية قابلة للاختبار، ثم حوكمتها عبر MCD قبل الفعل. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بتقليل أخطاء السببية، زيادة نجاح tool-use، وتحسن نقل المهارة بين مجالات لا تشترك في القاموس. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 1.3: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 1.5: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory

التعريف: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory. الأطروحة: الذكاء العام ليس اتساعا في الإجابة بل قدرة على تحويل الفروق Δ إلى مسارات سببية قابلة للاختبار، ثم حوكمتها عبر MCD قبل الفعل. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بتقليل أخطاء السببية، زيادة نجاح tool-use، وتحسن نقل المهارة بين مجالات لا تشترك في القاموس. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 1.5: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 1.6: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory

التعريف: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory. الأطروحة: الذكاء العام ليس اتساعا في الإجابة بل قدرة على تحويل الفروق Δ إلى مسارات سببية قابلة للاختبار، ثم حوكمتها عبر MCD قبل الفعل. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بتقليل أخطاء السببية، زيادة نجاح tool-use، وتحسن نقل المهارة بين مجالات لا تشترك في القاموس. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 1.6: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 1.7: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory

التعريف: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory. الأطروحة: الذكاء العام ليس اتساعا في الإجابة بل قدرة على تحويل الفروق Δ إلى مسارات سببية قابلة للاختبار، ثم حوكمتها عبر MCD قبل الفعل. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بتقليل أخطاء السببية، زيادة نجاح tool-use، وتحسن نقل المهارة بين مجالات لا تشترك في القاموس. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 1.7: نظرية الحاكمية الوعائية السببية Vascular Causal Governance Theory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 2.1: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory

التعريف: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory. الأطروحة: الحلقة agentic تصبح skill عندما تُضغط في إجراء قابل لإعادة الاستعمال، وتصبح hyperskill عندما تتحكم في اختيار وإصلاح وتقاعد المهارات الأخرى. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بأبلشن: إزالة طبقة hyperskill يجب أن تخفض transfer/recovery/library quality. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 2.1: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 2.2: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory

التعريف: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory. الأطروحة: الحلقة agentic تصبح skill عندما تُضغط في إجراء قابل لإعادة الاستعمال، وتصبح hyperskill عندما تتحكم في اختيار وإصلاح وتقاعد المهارات الأخرى. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بأبلشن: إزالة طبقة hyperskill يجب أن تخفض transfer/recovery/library quality. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 2.2: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 2.4: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory

التعريف: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory. الأطروحة: الحلقة agentic تصبح skill عندما تُضغط في إجراء قابل لإعادة الاستعمال، وتصبح hyperskill عندما تتحكم في اختيار وإصلاح وتقاعد المهارات الأخرى. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بأبلشن: إزالة طبقة hyperskill يجب أن تخفض transfer/recovery/library quality. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 2.4: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 2.5: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory

التعريف: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory. الأطروحة: الحلقة agentic تصبح skill عندما تُضغط في إجراء قابل لإعادة الاستعمال، وتصبح hyperskill عندما تتحكم في اختيار وإصلاح وتقاعد المهارات الأخرى. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بأبلشن: إزالة طبقة hyperskill يجب أن تخفض transfer/recovery/library quality. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 2.5: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 2.6: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory

التعريف: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory. الأطروحة: الحلقة agentic تصبح skill عندما تُضغط في إجراء قابل لإعادة الاستعمال، وتصبح hyperskill عندما تتحكم في اختيار وإصلاح وتقاعد المهارات الأخرى. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بأبلشن: إزالة طبقة hyperskill يجب أن تخفض transfer/recovery/library quality. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 2.6: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 2.7: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory

التعريف: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory. الأطروحة: الحلقة agentic تصبح skill عندما تُضغط في إجراء قابل لإعادة الاستعمال، وتصبح hyperskill عندما تتحكم في اختيار وإصلاح وتقاعد المهارات الأخرى. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بأبلشن: إزالة طبقة hyperskill يجب أن تخفض transfer/recovery/library quality. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 2.7: نظرية الهايبرسكل المساعفية Hyperskill Kernel Theory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 3.1: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance

التعريف: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance. الأطروحة: الإبداع التقني ليس فوضى أفكار؛ إنه رفع منضبط لاحتمالات البدائل ثم محكمة انتقاء تمنع الزخرفة وتبقي ما يملك آلية وقياسا. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بقياس novelty survivability بعد نقد خبراء وتحوير benchmarks. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 3.1: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 3.3: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance

التعريف: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance. الأطروحة: الإبداع التقني ليس فوضى أفكار؛ إنه رفع منضبط لاحتمالات البدائل ثم محكمة انتقاء تمنع الزخرفة وتبقي ما يملك آلية وقياسا. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بقياس novelty survivability بعد نقد خبراء وتحوير benchmarks. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 3.3: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 3.4: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance

التعريف: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance. الأطروحة: الإبداع التقني ليس فوضى أفكار؛ إنه رفع منضبط لاحتمالات البدائل ثم محكمة انتقاء تمنع الزخرفة وتبقي ما يملك آلية وقياسا. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بقياس novelty survivability بعد نقد خبراء وتحوير benchmarks. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 3.4: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 3.5: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance

التعريف: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance. الأطروحة: الإبداع التقني ليس فوضى أفكار؛ إنه رفع منضبط لاحتمالات البدائل ثم محكمة انتقاء تمنع الزخرفة وتبقي ما يملك آلية وقياسا. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بقياس novelty survivability بعد نقد خبراء وتحوير benchmarks. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 3.5: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب الخامس / النظرية 3.6: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance

التعريف: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance. الأطروحة: الإبداع التقني ليس فوضى أفكار؛ إنه رفع منضبط لاحتمالات البدائل ثم محكمة انتقاء تمنع الزخرفة وتبقي ما يملك آلية وقياسا. لا تقدم هذه النظرية كقانون مثبت بل كبنية توليدية تفتح أسئلة هندسية قابلة للقياس.

المقدمات: الذكاء العام يتطلب اختيارا بين مسارات لا مجرد إنتاج مخرجات؛ الفعل يحتاج ذاكرة ذات حدود؛ الأدوات تضيف قدرة وخطرا؛ والbenchmark يجب أن يقيس الاستقرار تحت التحوير لا الإجابة المحفوظة.

الآلية: تبدأ النظرية من negative space في الأدبيات. ما الغائب بين reasoning papers وagent benchmarks؟ الغائب هو طبقة تنتقي المسار وتتعلم من فشل المسارات نفسها. هنا تدخل الحاكمية الوعائية أو الهايبرسكل أو الإنتروبيا الاختراعية.

اختبارها: تُختبر بقياس novelty survivability بعد نقد خبراء وتحوير benchmarks. إذا لم يظهر فرق عند الأبلشن، تسقط النظرية إلى استعارة غير منتجة. وإذا ظهر فرق في مجال واحد فقط، تصبح نظرية محلية لا نظرية AGI عامة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث نظريات جديدة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الخامس / النظرية 3.6: نظرية حاكمية الإنتروبيا الاختراعية Inventive Entropy Governance نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم رتبي:** مرتبة الدليل: اختراع نموذجي مع آلية واختبار، لا إثبات تجريبي بعد.

الباب السادس / التكنولوجيا 1.1: MCAI-v2 Vascular Cortex OS

MCAI-v2 Vascular Cortex OS: نظام تشغيل معرفي يضع Difference Extractor وMCD Router وVascular Compute Allocator وPerturbation Court حول LLM. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: API: /difference, /mcd_route, /court, /world_effect, /memory_rewrite. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: court → verify → act → route → skill graph → causal map → evidence ranks → source ledger → ingest → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 1.1: MCAI-v2 Vascular Cortex OS نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بـtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 1.2: MCAI-v2 Vascular Cortex OS

MCAI-v2 Vascular Cortex OS: نظام تشغيل معرفي يضع Difference Extractor وMCD Router وVascular Compute Allocator وPerturbation Court حول LLM. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: API: /difference, /mcd_route, /court, /world_effect, /memory_rewrite. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 1.2: MCAI-v2 Vascular Cortex OS نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 1.3: MCAI-v2 Vascular Cortex OS

MCAI-v2 Vascular Cortex OS: نظام تشغيل معرفي يضع Difference Extractor وMCD Router وVascular Compute Allocator وPerturbation Court حول LLM. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: API: /difference, /mcd_route, /court, /world_effect, /memory_rewrite. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 1.3: MCAI-v2 Vascular Cortex OS نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 1.4: MCAI-v2 Vascular Cortex OS

MCAI-v2 Vascular Cortex OS: نظام تشغيل معرفي يضع Difference Extractor وMCD Router وVascular Compute Allocator وPerturbation Court حول LLM. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: API: /difference, /mcd_route, /court, /world_effect, /memory_rewrite. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 1.4: MCAI-v2 Vascular Cortex OS نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 1.5: MCAI-v2 Vascular Cortex OS

MCAI-v2 Vascular Cortex OS: نظام تشغيل معرفي يضع Difference Extractor وMCD Router وVascular Compute Allocator وPerturbation Court حول LLM. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: API: /difference, /mcd_route, /court, /world_effect, /memory_rewrite. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 1.5: MCAI-v2 Vascular Cortex OS نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 1.7: MCAI-v2 Vascular Cortex OS

MCAI-v2 Vascular Cortex OS: نظام تشغيل معرفي يضع Difference Extractor وMCD Router وVascular Compute Allocator وPerturbation Court حول LLM. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: API: /difference, /mcd_route, /court, /world_effect, /memory_rewrite. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 1.7: MCAI-v2 Vascular Cortex OS نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 2.1: Proof-Carrying HyperGraphRAG

Proof-Carrying HyperGraphRAG: GraphRAG لا يسترجع النص فقط بل يرجع عقدا بعلاقات وقيود ورتبة دليل وسجل اعتراضات. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: كل جواب يحمل مصدره، حدوده، مسار الاستدلال، ونقطة قابلة للهدم. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 2.1: Proof-Carrying HyperGraphRAG نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 2.2: Proof-Carrying HyperGraphRAG

Proof-Carrying HyperGraphRAG: GraphRAG لا يسترجع النص فقط بل يرجع عقدا بعلاقات وقيود ورتبة دليل وسجل اعتراضات. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: كل جواب يحمل مصدره، حدوده، مسار الاستدلال، ونقطة قابلة للهدم. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 2.2: Proof-Carrying HyperGraphRAG نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 2.3: Proof-Carrying HyperGraphRAG

Proof-Carrying HyperGraphRAG: GraphRAG لا يسترجع النص فقط بل يرجع عقدا بعلاقات وقيود ورتبة دليل وسجل اعتراضات. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: كل جواب يحمل مصدره، حدوده، مسار الاستدلال، ونقطة قابلة للهدم. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 2.3: Proof-Carrying HyperGraphRAG نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 2.5: Proof-Carrying HyperGraphRAG

Proof-Carrying HyperGraphRAG: GraphRAG لا يسترجع النص فقط بل يرجع عقدا بعلاقات وقيود ورتبة دليل وسجل اعتراضات. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: كل جواب يحمل مصدره، حدوده، مسار الاستدلال، ونقطة قابلة للهدم. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 2.5: Proof-Carrying HyperGraphRAG نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 2.6: Proof-Carrying HyperGraphRAG

Proof-Carrying HyperGraphRAG: GraphRAG لا يسترجع النص فقط بل يرجع عقدا بعلاقات وقيود ورتبة دليل وسجل اعتراضات. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: كل جواب يحمل مصدره، حدوده، مسار الاستدلال، ونقطة قابلة للهدم. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 2.6: Proof-Carrying HyperGraphRAG نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 2.7: Proof-Carrying HyperGraphRAG

Proof-Carrying HyperGraphRAG: GraphRAG لا يسترجع النص فقط بل يرجع عقدا بعلاقات وقيود ورتبة دليل وسجل اعتراضات. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: كل جواب يحمل مصدره، حدوده، مسار الاستدلال، ونقطة قابلة للهدم. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 2.7: Proof-Carrying HyperGraphRAG نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 3.1: Benchmark Mutation Foundry

Benchmark Mutation Foundry: مصنع يحول أي benchmark إلى عائلة اختبارات: عادية، معاكسة، توزيع جديد، أداة ملوثة، ذاكرة ناقصة، تكلفة محدودة. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: يعطي score للاستقرار لا للنجاح العارض فقط. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 3.1: Benchmark Mutation Foundry نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 3.2: Benchmark Mutation Foundry

Benchmark Mutation Foundry: مصنع يحول أي benchmark إلى عائلة اختبارات: عادية، معاكسة، توزيع جديد، أداة ملوثة، ذاكرة ناقصة، تكلفة محدودة. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: يعطي score للاستقرار لا للنجاح العارض فقط. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 3.2: Benchmark Mutation Foundry نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 3.4: Benchmark Mutation Foundry

Benchmark Mutation Foundry: مصنع يحول أي benchmark إلى عائلة اختبارات: عادية، معاكسة، توزيع جديد، أداة ملوثة، ذاكرة ناقصة، تكلفة محدودة. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: يعطي score للاستقرار لا للنجاح العارض فقط. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 3.4: Benchmark Mutation Foundry نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 3.5: Benchmark Mutation Foundry

Benchmark Mutation Foundry: مصنع يحول أي benchmark إلى عائلة اختبارات: عادية، معاكسة، توزيع جديد، أداة ملوثة، ذاكرة ناقصة، تكلفة محدودة. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: يعطي score للاستقرار لا للنجاح العارض فقط. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 3.5: Benchmark Mutation Foundry نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 3.6: Benchmark Mutation Foundry

Benchmark Mutation Foundry: مصنع يحول أي benchmark إلى عائلة اختبارات: عادية، معاكسة، توزيع جديد، أداة ملوثة، ذاكرة ناقصة، تكلفة محدودة. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: يعطي score للاستقرار لا للنجاح العارض فقط. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 3.6: Benchmark Mutation Foundry نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السادس / التكنولوجيا 3.7: Benchmark Mutation Foundry

Benchmark Mutation Foundry: مصنع يحول أي benchmark إلى عائلة اختبارات: عادية، معاكسة، توزيع جديد، أداة ملوثة، ذاكرة ناقصة، تكلفة محدودة. التقنية لا تقاس بجمال الاسم بل بعدد نقاط الفشل التي تكشفها وتقليلها. إذا لم تضف observability وrollback وmetrics فهي مجرد واجهة.

واجهة التنفيذ الأولية: يعطي score للاستقرار لا للنجاح العارض فقط. كل endpoint يجب أن يرجع trace لا نتيجة فقط: لماذا اختير المسار، ماذا رُفض، ما الدليل، ما الاعتراض، وماذا يجب أن يحدث لو تغير المصدر.

سلسلة التشغيل: ingest → source ledger → evidence ranks → causal map → skill graph → route → act → verify → court → memory. لا يسمح بتجاوز المحكمة في المسائل عالية المخاطر أو ذات claims قوية.

حدود التطبيق: التقنية ستكون بطيئة ومكلفة إن طبقت على كل سؤال. لذلك يتقدم MCD كآلية اقتصاد معرفي: الكلفة تصعد فقط عند العمق، الخطر، أو قيمة الأثر.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ثلاث تقنيات تنفيذية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السادس / التكنولوجيا 3.7: Benchmark Mutation Foundry نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **Prototype:** ابدأ بنسخة CLI: JSON in / JSON out، ثم أضف واجهة traces ولوحة benchmark mutation.

الباب السابع / أداة AGI تقنية 1: Δ-Extractor

Δ-Extractor: يستخرج الفروق النشطة: التناقض، النقص، التحول، اللايقين، المخاطر. الأداة لا تعيش مستقلة؛ تعيش داخل دورة MCAI-v2 حيث يحدد MCD متى تُستدعى ومتى تُمنع.

التنفيذ المقترح: مدخل: مسألة/مصدر. مخرج: vector للفروق مع رتبة خطر. يلزم تعريف schema واضح: input, output, confidence, evidence_rank, failure_modes, audit_trace. أي أداة بلا schema تتحول إلى prompt عائم.

مؤشرات الجودة: هل تخفض عدد الأخطاء؟ هل تزيد recoverability؟ هل تقلل كلفة التفكير؟ هل ترفع جودة المصادر؟ هل تجعل النقد أسهل؟ إن لم تحقق واحدة على الأقل، لا تستحق الوجود.

سيناريو فشل: قد تولد الأداة لغة مقنعة حول نفسها. العلاج: benchmark صغير عكسي يقيس الأداة ضد baseline بسيط. الاختراع لا يحمي نفسه باسمه بل بقياسه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب خمسة عشر أداة AGI تقنية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السابع / أداة AGI تقنية 1: Δ-Extractor نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **توصيل داخل النظام:** Δ-Extractor ↔ MCD Router ↔ Perturbation Court ↔ Self-Reconstruction Memory.

الباب السابع / أداة AGI تقنية 3: Vascular Compute Allocator

Vascular Compute Allocator: يوزع الميزانية بين بحث، تفكير، أدوات، محاكاة، محكمة نقد. الأداة لا تعيش مستقلة؛ تعيش داخل دورة MCAI-v2 حيث يحدد MCD متى تُستدعى ومتى تُمنع.

التنفيذ المقترح: يقيس marginal value of compute لكل مسار. يلزم تعريف schema واضح: input, output, confidence, evidence_rank, failure_modes, audit_trace. أي أداة بلا schema تتحول إلى prompt عائم.

مؤشرات الجودة: هل تخفض عدد الأخطاء؟ هل تزيد recoverability؟ هل تقلل كلفة التفكير؟ هل ترفع جودة المصادر؟ هل تجعل النقد أسهل؟ إن لم تحقق واحدة على الأقل، لا تستحق الوجود.

سيناريو فشل: قد تولد الأداة لغة مقنعة حول نفسها. العلاج: benchmark صغير عكسي يقيس الأداة ضد baseline بسيط. الاختراع لا يحمي نفسه باسمه بل بقياسه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب خمسة عشر أداة AGI تقنية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السابع / أداة AGI تقنية 3: Vascular Compute Allocator نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **توصيل داخل النظام:** Vascular Compute Allocator ↔ MCD Router ↔ Perturbation Court ↔ Self-Reconstruction ↔ Memory.

الباب السابع / أداة AGI تقنية 4: Evidence Rank Matrix

Evidence Rank Matrix: يفصل الثابت/الراجح/الخلافي/الفرضية/الاختراع/التأمل. الأداة لا تعيش مستقلة؛ تعيش داخل دورة MCAI-v2 حيث يحدد MCD متى تُستدعى ومتى تُمنع.

التنفيذ المقترح: يمنع خلط الحماس العلمي بالنتيجة المثبتة. يلزم تعريف schema واضح: input, output, confidence, evidence_rank, failure_modes, audit_trace. أي أداة بلا schema تتحول إلى prompt عائم.

مؤشرات الجودة: هل تخفض عدد الأخطاء؟ هل تزيد recoverability؟ هل تقلل كلفة التفكير؟ هل ترفع جودة المصادر؟ هل تجعل النقد أسهل؟ إن لم تحقق واحدة على الأقل، لا تستحق الوجود.

سيناريو فشل: قد تولد الأداة لغة مقنعة حول نفسها. العلاج: benchmark صغير عكسي يقيس الأداة ضد baseline بسيط. الاختراع لا يحمي نفسه باسمه بل بقياسه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب خمسة عشر أداة AGI تقنية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السابع / أداة AGI تقنية 4: Evidence Rank Matrix نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **توصيل داخل النظام:** Evidence Rank Matrix ↔ MCD Router ↔ Perturbation Court ↔ Self-Reconstruction ↔ Memory.

الباب السابع / أداة AGI تقنية 5: Causal Map Compiler

Causal Map Compiler: يحول النص إلى causal graph: عقد، سهام، قيود، استثناءات. الأداة لا تعيش مستقلة؛ تعيش داخل دورة MCAI-v2 حيث يحدد MCD متى تُستدعى ومتى تُمنع.

التنفيذ المقترح: يفضح العلاقات الزائفة والسببية غير المبرهنة. يلزم تعريف schema واضح: input, output, confidence, evidence_rank, failure_modes, audit_trace. أي أداة بلا schema تتحول إلى prompt عائم.

مؤشرات الجودة: هل تخفض عدد الأخطاء؟ هل تزيد recoverability؟ هل تقلل كلفة التفكير؟ هل ترفع جودة المصادر؟ هل تجعل النقد أسهل؟ إن لم تحقق واحدة على الأقل، لا تستحق الوجود.

سيناريو فشل: قد تولد الأداة لغة مقنعة حول نفسها. العلاج: benchmark صغير عكسي يقيس الأداة ضد baseline بسيط. الاختراع لا يحمي نفسه باسمه بل بقياسه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب خمسة عشر أداة AGI تقنية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السابع / أداة AGI تقنية 5: Causal Map Compiler نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **توصيل داخل النظام:** Causal Map Compiler ↔ MCD Router ↔ Perturbation Court ↔ Self-Reconstruction ↔ Memory.

الباب السابع / أداة AGI تقنية 6: Hypothesis Reactor

Hypothesis Reactor: يولد فرضيات من negative space وinversion وcross-domain fusion. الأداة لا تعيش مستقلة؛ تعيش داخل دورة MCAI-v2 حيث يحدد MCD متى تُستدعى ومتى تُمنع.

التنفيذ المقترح: كل فرضية تمر عبر فهم/آلية/إضافة. يلزم تعريف schema واضح: input, output, confidence, evidence_rank, failure_modes, audit_trace. أي أداة بلا schema تتحول إلى prompt عائم.

مؤشرات الجودة: هل تخفض عدد الأخطاء؟ هل تزيد recoverability؟ هل تقلل كلفة التفكير؟ هل ترفع جودة المصادر؟ هل تجعل النقد أسهل؟ إن لم تحقق واحدة على الأقل، لا تستحق الوجود.

سيناريو فشل: قد تولد الأداة لغة مقنعة حول نفسها. العلاج: benchmark صغير عكسي يقيس الأداة ضد baseline بسيط. الاختراع لا يحمي نفسه باسمه بل بقياسه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب خمسة عشر أداة AGI تقنية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السابع / أداة AGI تقنية 6: Hypothesis Reactor نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **توصيل داخل النظام:** Hypothesis Reactor ↔ MCD Router ↔ Perturbation Court ↔ Self-Reconstruction Memory.

الباب السابع / أداة AGI تقنية 8: Self-Reconstruction Memory

Self-Reconstruction Memory: يسجل أسباب النجاح والفشل لا النتائج فقط. الأداة لا تعيش مستقلة؛ تعيش داخل دورة MCAI-v2 حيث يحدد MCD متى تُستدعى ومتى تُمنع.

التنفيذ المقترح: ذاكرة حاكمة لا ذاكرة ملوثة بالانطباعات. يلزم تعريف schema واضح: input, output, confidence, evidence_rank, failure_modes, audit_trace. أي أداة بلا schema تتحول إلى prompt عائم.

مؤشرات الجودة: هل تخفض عدد الأخطاء؟ هل تزيد recoverability؟ هل تقلل كلفة التفكير؟ هل ترفع جودة المصادر؟ هل تجعل النقد أسهل؟ إن لم تحقق واحدة على الأقل، لا تستحق الوجود.

سيناريو فشل: قد تولد الأداة لغة مقنعة حول نفسها. العلاج: benchmark صغير عكسي يقيس الأداة ضد baseline بسيط. الاختراع لا يحمي نفسه باسمه بل بقياسه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب خمسة عشر أداة AGI تقنية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السابع / أداة AGI تقنية 8: Self-Reconstruction Memory نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **توصيل داخل النظام:** Self-Reconstruction Memory ↔ MCD Router ↔ Perturbation Court ↔ Self-Reconstruction Memory.

الباب السابع / أداة AGI تقنية 9: World-Effect Loop

World-Effect Loop: يربط الجواب بأثر في العالم: كود، تجربة، تعديل، قرار، قياس. الأداة لا تعيش مستقلة؛ تعيش داخل دورة MCAI-v2 حيث يحدد MCD متى تُستدعى ومتى تُمنع.

التنفيذ المقترح: يعالج انفصال اللغة عن الفعل. يلزم تعريف schema واضح: input, output, confidence, evidence_rank, failure_modes, audit_trace. أي أداة بلا schema تتحول إلى prompt عائم.

مؤشرات الجودة: هل تخفض عدد الأخطاء؟ هل تزيد recoverability؟ هل تقلل كلفة التفكير؟ هل ترفع جودة المصادر؟ هل تجعل النقد أسهل؟ إن لم تحقق واحدة على الأقل، لا تستحق الوجود.

سيناريو فشل: قد تولد الأداة لغة مقنعة حول نفسها. العلاج: benchmark صغير عكسي يقيس الأداة ضد baseline بسيط. الاختراع لا يحمي نفسه باسمه بل بقياسه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب خمسة عشر أداة AGI تقنية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السابع / أداة AGI تقنية 9: World-Effect Loop نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **توصيل داخل النظام:** World-Effect Loop ↔ MCD Router ↔ Perturbation Court ↔ Self-Reconstruction Memory.

الباب السابع / أداة AGI تقنية 10: Skill Graph Compiler

Skill Graph Compiler: يبني شبكة مهارات: شروط، بدائل، تعارضات، Verifiers. الأداة لا تعيش مستقلة؛ تعيش داخل دورة MCAI-v2 حيث يحدد MCD متى تُستدعى ومتى تُمنع.

التنفيذ المقترح: يحول agent loop إلى مكتبة قابلة للتطور. يلزم تعريف schema واضح: input, output, confidence, evidence_rank, failure_modes, audit_trace. أي أداة بلا schema تتحول إلى prompt عائم.

مؤشرات الجودة: هل تخفض عدد الأخطاء؟ هل تزيد recoverability؟ هل تقلل كلفة التفكير؟ هل ترفع جودة المصادر؟ هل تجعل النقد أسهل؟ إن لم تحقق واحدة على الأقل، لا تستحق الوجود.

سيناريو فشل: قد تولد الأداة لغة مقنعة حول نفسها. العلاج: benchmark صغير عكسي يقيس الأداة ضد baseline بسيط. الاختراع لا يحمي نفسه باسمه بل بقياسه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب خمسة عشر أداة AGI تقنية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السابع / أداة AGI تقنية 10: Skill Graph Compiler نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **توصيل داخل النظام:** Skill Graph Compiler ↔ MCD Router ↔ Perturbation Court ↔ Self-Reconstruction ↔ Memory.

الباب السابع / أداة AGI تقنية 11: Negative-Space Miner

Negative-Space Miner: يسأل ما الذي لم تقله المصادر لأنه بين التخصصات. الأداة لا تعيش مستقلة؛ تعيش داخل دورة MCAI-v2 حيث يحدد MCD متى تُستدعى ومتى تُمنع.

التنفيذ المقترح: مصدر رئيسي للاختراع المنضبط. يلزم تعريف schema واضح: input, output, confidence, evidence_rank, failure_modes, audit_trace. أي أداة بلا schema تتحول إلى prompt عائم.

مؤشرات الجودة: هل تخفض عدد الأخطاء؟ هل تزيد recoverability؟ هل تقلل كلفة التفكير؟ هل ترفع جودة المصادر؟ هل تجعل النقد أسهل؟ إن لم تحقق واحدة على الأقل، لا تستحق الوجود.

سيناريو فشل: قد تولد الأداة لغة مقنعة حول نفسها. العلاج: benchmark صغير عكسي يقيس الأداة ضد baseline بسيط. الاختراع لا يحمي نفسه باسمه بل بقياسه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب خمسة عشر أداة AGI تقنية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السابع / أداة AGI تقنية 11: Negative-Space Miner نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **توصيل داخل النظام:** Negative-Space Miner ↔ MCD Router ↔ Perturbation Court ↔ Self-Reconstruction Memory.

الباب السابع / أداة AGI تقنية 12: Tool-Risk Firewall

Tool-Risk Firewall: يصنف الأدوات حسب الصلاحيات، قابلية الحقن، كلفة الخطأ، rollback. الأداة لا تعيش مستقلة؛ تعيش داخل دورة MCAI-v2 حيث يحدد MCD متى تُستدعى ومتى تُمنع.

التنفيذ المقترح: ضروري مع MCP وcomputer use. يلزم تعريف schema واضح: input, output, confidence, evidence_rank, failure_modes, audit_trace. أي أداة بلا schema تتحول إلى prompt عائم.

مؤشرات الجودة: هل تخفض عدد الأخطاء؟ هل تزيد recoverability؟ هل تقلل كلفة التفكير؟ هل ترفع جودة المصادر؟ هل تجعل النقد أسهل؟ إن لم تحقق واحدة على الأقل، لا تستحق الوجود.

سيناريو فشل: قد تولد الأداة لغة مقنعة حول نفسها. العلاج: benchmark صغير عكسي يقيس الأداة ضد baseline بسيط. الاختراع لا يحمي نفسه باسمه بل بقياسه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب خمسة عشر أداة AGI تقنية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السابع / أداة AGI تقنية 12: Tool-Risk Firewall نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **توصيل داخل النظام:** Tool-Risk Firewall ↔ MCD Router ↔ Perturbation Court ↔ Self-Reconstruction Memory.

الباب السابع / أداة AGI تقنية 14: Failure Atlas

Failure Atlas: يفهرس الفشل: perception, state, planning, tool, verification, memory, stop. الأداة لا تعيش مستقلة؛ تعيش داخل دورة MCAI-v2 حيث يحدد MCD متى تُستدعى ومتى تُمنع.

التنفيذ المقترح: يحسن النظام بتشخيص لا بلوم. يلزم تعريف schema واضح: input, output, confidence, evidence_rank, failure_modes, audit_trace. أي أداة بلا schema تتحول إلى prompt عائم.

مؤشرات الجودة: هل تخفض عدد الأخطاء؟ هل تزيد recoverability؟ هل تقلل كلفة التفكير؟ هل ترفع جودة المصادر؟ هل تجعل النقد أسهل؟ إن لم تحقق واحدة على الأقل، لا تستحق الوجود.

سيناريو فشل: قد تولد الأداة لغة مقنعة حول نفسها. العلاج: benchmark صغير عكسي يقيس الأداة ضد baseline بسيط. الاختراع لا يحمي نفسه باسمه بل بقياسه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب خمسة عشر أداة AGI تقنية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السابع / أداة AGI تقنية 14: Failure Atlas نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **توصيل داخل النظام:** Failure Atlas ↔ MCD Router ↔ Perturbation Court ↔ Self-Reconstruction Memory.

الباب السابع / أداة AGI تقنية 15: Novelty Survivability Scorer

Novelty Survivability Scorer: يقيم الفكرة الجديدة بعد محكمة نقد ومحاولات اختزال. الأداة لا تعيش مستقلة؛ تعيش داخل دورة MCAI-v2 حيث يحدد MCD متى تُستدعى ومتى تُمنع.

التنفيذ المقترح: يفرق بين الاختراع والزخرفة. يلزم تعريف schema واضح: input, output, confidence, evidence_rank, failure_modes, audit_trace. أي أداة بلا schema تتحول إلى prompt عائم.

مؤشرات الجودة: هل تخفض عدد الأخطاء؟ هل تزيد recoverability؟ هل تقلل كلفة التفكير؟ هل ترفع جودة المصادر؟ هل تجعل النقد أسهل؟ إن لم تحقق واحدة على الأقل، لا تستحق الوجود.

سيناريو فشل: قد تولد الأداة لغة مقنعة حول نفسها. العلاج: benchmark صغير عكسي يقيس الأداة ضد baseline بسيط. الاختراع لا يحمي نفسه باسمه بل بقياسه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب خمسة عشر أداة AGI تقنية لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب السابع / أداة AGI تقنية 15: Novelty Survivability Scorer نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **توصيل داخل النظام:** Novelty Survivability Scorer ↔ MCD Router ↔ Perturbation Court ↔ Self-Reconstruction Memory.

الباب الثامن / فلسفة تدريب 1.1: التدريب على الحاكمية لا الإجابة

التدريب على الحاكمية لا الإجابة: المطلوب أن يعرف النموذج متى يبحث، متى يشك، متى يوقف، ومتى يصعد الصلاحية. المقصود ليس تدريب نموذج على style مساعفي، بل تدريب نظام على قرارات حاكمية قابلة للقياس.

البيانات المطلوبة ليست إجابات فقط. نحتاج traces: مسارات رُفضت، أدوات مُنعت، مصادر سقطت، اعتراضات حسنت جوابا، ذاكرة لم تُكتب لأنها ملوثة. هكذا يتعلم النظام من قرارات عدم الفعل كما يتعلم من الفعل.

التمييز الحاسم: fine-tuning يحسن استجابة داخل توزيع، أما MCAI training فيحسن protocol selection داخل اختلافات التوزيع. لذلك نحتاج datasets تسجل سياق الاختيار لا النتيجة النهائية فقط.

اختبار الجودة: ablation على طبقة التدريب. إذا أزلنا traces الفشل أو أدوات الحوكمة وانخفضت robustness في benchmarks المتحورة، فهذا دليل تشغيل لا برهان نهائي.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / فلسفة تدريب 1.1: التدريب على الحاكمية لا الإجابة نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة تدريب:** درّب على متى لا تجيب، متى تبحث، متى تشك، متى تستخدم الأداة، ومتى تمحو الذاكرة.

الباب الثامن / فلسفة تدريب 1.2: التدريب على الحاكمية لا الإجابة

التدريب على الحاكمية لا الإجابة: المطلوب أن يعرف النموذج متى يبحث، متى يشك، متى يوقف، ومتى يصعد الصلاحية. المقصود ليس تدريب نموذج على style مساعفي، بل تدريب نظام على قرارات حاكمية قابلة للقياس.

البيانات المطلوبة ليست إجابات فقط. نحتاج traces: مسارات رُفضت، أدوات مُنعت، مصادر سقطت، اعتراضات حسنت جوابا، ذاكرة لم تُكتب لأنها ملوثة. هكذا يتعلم النظام من قرارات عدم الفعل كما يتعلم من الفعل.

التمييز الحاسم: fine-tuning يحسن استجابة داخل توزيع، أما MCAI training فيحسن protocol selection داخل اختلافات التوزيع. لذلك نحتاج datasets تسجل سياق الاختيار لا النتيجة النهائية فقط.

اختبار الجودة: ablation على طبقة التدريب. إذا أزلنا traces الفشل أو أدوات الحوكمة وانخفضت robustness في benchmarks المتحورة، فهذا دليل تشغيل لا برهان نهائي.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / فلسفة تدريب 1.2: التدريب على الحاكمية لا الإجابة نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة تدريب:** درّب على متى لا تجيب، متى تبحث، متى تشك، متى تستخدم الأداة، ومتى تمحو الذاكرة.

الباب الثامن / فلسفة تدريب 2.2: التدريب على ذاكرة الفشل

التدريب على ذاكرة الفشل: كل failure trace يتحول إلى قاعدة مستقبلية لا إلى نسيان. المقصود ليس تدريب نموذج على style مساعفي، بل تدريب نظام على قرارات حاكمية قابلة للقياس.

البيانات المطلوبة ليست إجابات فقط. نحتاج traces: مسارات رُفضت، أدوات مُنعت، مصادر سقطت، اعتراضات حسنت جوابا، ذاكرة لم تُكتب لأنها ملوثة. هكذا يتعلم النظام من قرارات عدم الفعل كما يتعلم من الفعل.

التمييز الحاسم: fine-tuning يحسن استجابة داخل توزيع، أما MCAI training فيحسن protocol selection داخل اختلافات التوزيع. لذلك نحتاج datasets تسجل سياق الاختيار لا النتيجة النهائية فقط.

اختبار الجودة: ablation على طبقة التدريب. إذا أزلنا traces الفشل أو أدوات الحوكمة وانخفضت robustness في benchmarks المتحورة، فهذا دليل تشغيل لا برهان نهائي.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / فلسفة تدريب 2.2: التدريب على ذاكرة الفشل نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة تدريب:** درّب على متى لا تجيب، متى تبحث، متى تشك، متى تستخدم الأداة، ومتى تمحو الذاكرة.

الباب الثامن / فلسفة تدريب 3.1: التدريب على السببية والأثر

التدريب على السببية والأثر: لا يكفي النص؛ يجب ربطه بأثر قابل للرصد. المقصود ليس تدريب نموذج على style مساعفي، بل تدريب نظام على قرارات حاكمية قابلة للقياس.

البيانات المطلوبة ليست إجابات فقط. نحتاج traces: مسارات رُفضت، أدوات مُنعت، مصادر سقطت، اعتراضات حسنت جوابا، ذاكرة لم تُكتب لأنها ملوثة. هكذا يتعلم النظام من قرارات عدم الفعل كما يتعلم من الفعل.

التمييز الحاسم: fine-tuning يحسن استجابة داخل توزيع، أما MCAI training فيحسن protocol selection داخل اختلافات التوزيع. لذلك نحتاج datasets تسجل سياق الاختيار لا النتيجة النهائية فقط.

اختبار الجودة: ablation على طبقة التدريب. إذا أزلنا traces الفشل أو أدوات الحوكمة وانخفضت robustness في benchmarks المتحورة، فهذا دليل تشغيل لا برهان نهائي.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / فلسفة تدريب 3.1: التدريب على السببية والأثر نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة تدريب:** درّب على متى لا تجيب، متى تبحث، متى تشك، متى تستخدم الأداة، ومتى تمحو الذاكرة.

الباب الثامن / فلسفة تدريب 3.2: التدريب على السببية والأثر

التدريب على السببية والأثر: لا يكفي النص؛ يجب ربطه بأثر قابل للرصد. المقصود ليس تدريب نموذج على style مساعفي، بل تدريب نظام على قرارات حاكمية قابلة للقياس.

البيانات المطلوبة ليست إجابات فقط. نحتاج traces: مسارات رُفضت، أدوات مُنعت، مصادر سقطت، اعتراضات حسنت جوابا، ذاكرة لم تُكتب لأنها ملوثة. هكذا يتعلم النظام من قرارات عدم الفعل كما يتعلم من الفعل.

التمييز الحاسم: fine-tuning يحسن استجابة داخل توزيع، أما MCAI training فيحسن protocol selection داخل اختلافات التوزيع. لذلك نحتاج datasets تسجل سياق الاختيار لا النتيجة النهائية فقط.

اختبار الجودة: ablation على طبقة التدريب. إذا أزلنا traces الفشل أو أدوات الحوكمة وانخفضت robustness في benchmarks المتحورة، فهذا دليل تشغيل لا برهان نهائي.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / فلسفة تدريب 3.2: التدريب على السببية والأثر نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة تدريب:** درّب على متى لا تجيب، متى تبحث، متى تشك، متى تستخدم الأداة، ومتى تمحو الذاكرة.

الباب الثامن / فلسفة تدريب 4.1: التدريب على الانتقاء الاختراعي

التدريب على الانتقاء الاختراعي: توليد بدائل كثيرة ثم انتقاء ما يملك آلية ودليلا. المقصود ليس تدريب نموذج على style مساعفي، بل تدريب نظام على قرارات حاكمية قابلة للقياس.

البيانات المطلوبة ليست إجابات فقط. نحتاج traces: مسارات رُفضت، أدوات مُنعت، مصادر سقطت، اعتراضات حسنت جوابا، ذاكرة لم تُكتب لأنها ملوثة. هكذا يتعلم النظام من قرارات عدم الفعل كما يتعلم من الفعل.

التمييز الحاسم: fine-tuning يحسن استجابة داخل توزيع، أما MCAI training فيحسن protocol selection داخل اختلافات التوزيع. لذلك نحتاج datasets تسجل سياق الاختيار لا النتيجة النهائية فقط.

اختبار الجودة: ablation على طبقة التدريب. إذا أزلنا traces الفشل أو أدوات الحوكمة وانخفضت robustness في benchmarks المتحورة، فهذا دليل تشغيل لا برهان نهائي.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / فلسفة تدريب 4.1: التدريب على الانتقاء الاختراعي نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة تدريب:** درّب على متى لا تجيب، متى تبحث، متى تشك، متى تستخدم الأداة، ومتى تمحو الذاكرة.

الباب الثامن / فلسفة تدريب 4.2: التدريب على الانتقاء الاختراعي

التدريب على الانتقاء الاختراعي: توليد بدائل كثيرة ثم انتقاء ما يملك آلية ودليلا. المقصود ليس تدريب نموذج على style مساعفي، بل تدريب نظام على قرارات حاكمية قابلة للقياس.

البيانات المطلوبة ليست إجابات فقط. نحتاج traces: مسارات رُفضت، أدوات مُنعت، مصادر سقطت، اعتراضات حسنت جوابا، ذاكرة لم تُكتب لأنها ملوثة. هكذا يتعلم النظام من قرارات عدم الفعل كما يتعلم من الفعل.

التمييز الحاسم: fine-tuning يحسن استجابة داخل توزيع، أما MCAI training فيحسن protocol selection داخل اختلافات التوزيع. لذلك نحتاج datasets تسجل سياق الاختيار لا النتيجة النهائية فقط.

اختبار الجودة: ablation على طبقة التدريب. إذا أزلنا traces الفشل أو أدوات الحوكمة وانخفضت robustness في benchmarks المتحورة، فهذا دليل تشغيل لا برهان نهائي.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / فلسفة تدريب 4.2: التدريب على الانتقاء الاختراعي نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة تدريب:** درّب على متى لا تجيب، متى تبحث، متى تشك، متى تستخدم الأداة، ومتى تمحو الذاكرة.

الباب الثامن / فلسفة تدريب 5.2: التدريب على تواضع معرفي هجومي

التدريب على تواضع معرفي هجومي: نهاجم المجال بقوة لكن نضع كل ادعاء في مرتبته. المقصود ليس تدريب نموذج على style مساعفي، بل تدريب نظام على قرارات حاكمية قابلة للقياس.

البيانات المطلوبة ليست إجابات فقط. نحتاج traces: مسارات رُفضت، أدوات مُنعت، مصادر سقطت، اعتراضات حسنت جوابا، ذاكرة لم تُكتب لأنها ملوثة. هكذا يتعلم النظام من قرارات عدم الفعل كما يتعلم من الفعل.

التمييز الحاسم: fine-tuning يحسن استجابة داخل توزيع، أما MCAI training فيحسن protocol selection داخل اختلافات التوزيع. لذلك نحتاج datasets تسجل سياق الاختيار لا النتيجة النهائية فقط.

اختبار الجودة: ablation على طبقة التدريب. إذا أزلنا traces الفشل أو أدوات الحوكمة وانخفضت robustness في benchmarks المتحورة، فهذا دليل تشغيل لا برهان نهائي.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / فلسفة تدريب 5.2: التدريب على تواضع معرفي هجومي نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة تدريب:** درّب على متى لا تجيب، متى تبحث، متى تشك، متى تستخدم الأداة، ومتى تمحو الذاكرة.

الباب الثامن / أداة تدريب 1.1: MCD Curriculum Generator

MCD Curriculum Generator: يبني منهجا تصاعديا حسب فرق التعقيد لا حسب ترتيب كتاب مدرسي. هذه الأداة تنتج مواد تدريب للنظام الحاكمي لا مجرد أمثلة سؤال/جواب.

بنية البيانات: task_id, initial_difference, MCD_vector, selected_route, rejected_routes, tools_allowed, tools_denied, verifier_result, court_attacks, final_effect, memory_write. هذه الحقول تحول التجربة إلى معرفة قابلة للاسترجاع.

مؤشر النجاح: لا نقيس الدقة فقط بل الاستقرار عبر تغييرات صغيرة: مصدر ناقص، أداة مضللة، قيد أمني، توزيع جديد، سياق أطول، أو سؤال يحتوي نخا إبستمولوجيا.

الخطر: dataset قد يعلّم النظام طاعة شكلية للمحكمة لا نقدا حقيقيا. العلاج: محكمة خارجية تولد اعتراضات غير مكررة، وقياس novelty survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / أداة تدريب 1.1: MCD Curriculum Generator نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مخرج عملي:** ملف JSONL لتدريب/تقييم loop governance، لا مقال تنظيري.

الباب الثامن / أداة تدريب 1.2: MCD Curriculum Generator

MCD Curriculum Generator: يبني منهجا تصاعديا حسب فرق التعقيد لا حسب ترتيب كتاب مدرسي. هذه الأداة تنتج مواد تدريب للنظام الحاكمي لا مجرد أمثلة سؤال/جواب.

بنية البيانات: task_id, initial_difference, MCD_vector, selected_route, rejected_routes, tools_allowed, tools_denied, verifier_result, court_attacks, final_effect, memory_write. هذه الحقول تحول التجربة إلى معرفة قابلة للاسترجاع.

مؤشر النجاح: لا نقيس الدقة فقط بل الاستقرار عبر تغييرات صغيرة: مصدر ناقص، أداة مضللة، قيد أمني، توزيع جديد، سياق أطول، أو سؤال يحتوي نخا إبستمولوجيا.

الخطر: dataset قد يعلّم النظام طاعة شكلية للمحكمة لا نقدا حقيقيا. العلاج: محكمة خارجية تولد اعتراضات غير مكررة، وقياس novelty survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / أداة تدريب 1.2: MCD Curriculum Generator نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مخرج عملي:** ملف JSONL لتدريب/تقييم loop governance، لا مقال تنظيري.

الباب الثامن / أداة تدريب 2.1: Refutation Trajectory Dataset

Refutation Trajectory Dataset: مجموعة بيانات لسلاسل نقد وإصلاح لا لسلاسل إجابة فقط. هذه الأداة تنتج مواد تدريب للنظام الحاكمي لا مجرد أمثلة سؤال/جواب.

بنية البيانات: task_id, initial_difference, MCD_vector, selected_route, rejected_routes, tools_allowed, tools_denied, verifier_result, court_attacks, final_effect, memory_write. هذه الحقول تحول التجربة إلى معرفة قابلة للاسترجاع.

مؤشر النجاح: لا نقيس الدقة فقط بل الاستقرار عبر تغييرات صغيرة: مصدر ناقص، أداة مضللة، قيد أمني، توزيع جديد، سياق أطول، أو سؤال يحتوي نخا إبستمولوجيا.

الخطر: dataset قد يعلّم النظام طاعة شكلية للمحكمة لا نقدا حقيقيا. العلاج: محكمة خارجية تولد اعتراضات غير مكررة، وقياس novelty survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / أداة تدريب 2.1: Refutation Trajectory Dataset نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بـtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مخرج عملي:** ملف JSONL لتدريب/تقييم loop governance، لا مقال تنظيري.

الباب الثامن / أداة تدريب 3.1: Causal Tool-Use Traces

Causal Tool-Use Traces: أمثلة تربط tool call بسبب، بديل، تحقق، rollback. هذه الأداة تنتج مواد تدريب للنظام الحاكمي لا مجرد أمثلة سؤال/جواب.

بنية البيانات: task_id, initial_difference, MCD_vector, selected_route, rejected_routes, tools_allowed, tools_denied, verifier_result, court_attacks, final_effect, memory_write. هذه الحقول تحول التجربة إلى معرفة قابلة للاسترجاع.

مؤشر النجاح: لا نقيس الدقة فقط بل الاستقرار عبر تغييرات صغيرة: مصدر ناقص، أداة مضللة، قيد أمني، توزيع جديد، سياق أطول، أو سؤال يحتوي نخا إبستمولوجيا.

الخطر: dataset قد يعلّم النظام طاعة شكلية للمحكمة لا نقدا حقيقيا. العلاج: محكمة خارجية تولد اعتراضات غير مكررة، وقياس novelty survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / أداة تدريب 3.1: Causal Tool-Use Traces نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بـtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مخرج عملي:** ملف JSONL لتدريب/تقييم loop governance، لا مقال تنظيري.

الباب الثامن / أداة تدريب 3.2: Causal Tool-Use Traces

Causal Tool-Use Traces: أمثلة تربط tool call بسبب، بديل، تحقق، rollback. هذه الأداة تنتج مواد تدريب للنظام الحاكمي لا مجرد أمثلة سؤال/جواب.

بنية البيانات: task_id, initial_difference, MCD_vector, selected_route, rejected_routes, tools_allowed, tools_denied, verifier_result, court_attacks, final_effect, memory_write. هذه الحقول تحول التجربة إلى معرفة قابلة للاسترجاع.

مؤشر النجاح: لا نقيس الدقة فقط بل الاستقرار عبر تغييرات صغيرة: مصدر ناقص، أداة مضللة، قيد أمني، توزيع جديد، سياق أطول، أو سؤال يحتوي نخا إبستمولوجيا.

الخطر: dataset قد يعلّم النظام طاعة شكلية للمحكمة لا نقدا حقيقيا. العلاج: محكمة خارجية تولد اعتراضات غير مكررة، وقياس novelty survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / أداة تدريب 3.2: Causal Tool-Use Traces نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مخرج عملي:** ملف JSONL لتدريب/تقييم loop governance، لا مقال تنظيري.

الباب الثامن / أداة تدريب 4.1: Skill-Graph Distillation

Skill-Graph Distillation: تقطير المسارات الناجحة إلى مهارات قابلة لإعادة الاستخدام. هذه الأداة تنتج مواد تدريب للنظام الحاكمي لا مجرد أمثلة سؤال/جواب.

بنية البيانات: task_id, initial_difference, MCD_vector, selected_route, rejected_routes, tools_allowed, tools_denied, verifier_result, court_attacks, final_effect, memory_write. هذه الحقول تحول التجربة إلى معرفة قابلة للاسترجاع.

مؤشر النجاح: لا نقيس الدقة فقط بل الاستقرار عبر تغييرات صغيرة: مصدر ناقص، أداة مضللة، قيد أمني، توزيع جديد، سياق أطول، أو سؤال يحتوي نخا إبستمولوجيا.

الخطر: dataset قد يعلّم النظام طاعة شكلية للمحكمة لا نقدا حقيقيا. العلاج: محكمة خارجية تولد اعتراضات غير مكررة، وقياس novelty survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / أداة تدريب 4.1: Skill-Graph Distillation نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مخرج عملي:** ملف JSONL لتدريب/تقييم loop governance، لا مقال تنظيري.

الباب الثامن / أداة تدريب 4.2: Skill-Graph Distillation

Skill-Graph Distillation: تقطير المسارات الناجحة إلى مهارات قابلة لإعادة الاستخدام. هذه الأداة تنتج مواد تدريب للنظام الحاكمي لا مجرد أمثلة سؤال/جواب.

بنية البيانات: task_id, initial_difference, MCD_vector, selected_route, rejected_routes, tools_allowed, tools_denied, verifier_result, court_attacks, final_effect, memory_write. هذه الحقول تحول التجربة إلى معرفة قابلة للاسترجاع.

مؤشر النجاح: لا نقيس الدقة فقط بل الاستقرار عبر تغييرات صغيرة: مصدر ناقص، أداة مضللة، قيد أمني، توزيع جديد، سياق أطول، أو سؤال يحتوي نخا إبستمولوجيا.

الخطر: dataset قد يعلّم النظام طاعة شكلية للمحكمة لا نقدا حقيقيا. العلاج: محكمة خارجية تولد اعتراضات غير مكررة، وقياس novelty survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / أداة تدريب 4.2: Skill-Graph Distillation نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مخرج عملي:** ملف JSONL لتدريب/تقييم loop governance، لا مقال تنظيري.

الباب الثامن / أداة تدريب 5.1: World-Effect Regression Suite

World-Effect Regression Suite: اختبارات تضمن أن الإخراج يحدث أثرا صحيحا لا لغة جميلة فقط. هذه الأداة تنتج مواد تدريب للنظام الحاكمي لا مجرد أمثلة سؤال/جواب.

بنية البيانات: task_id, initial_difference, MCD_vector, selected_route, rejected_routes, tools_allowed, tools_denied, verifier_result, court_attacks, final_effect, memory_write. هذه الحقول تحول التجربة إلى معرفة قابلة للاسترجاع.

مؤشر النجاح: لا نقيس الدقة فقط بل الاستقرار عبر تغييرات صغيرة: مصدر ناقص، أداة مضللة، قيد أمني، توزيع جديد، سياق أطول، أو سؤال يحتوي نخا إبستمولوجيا.

الخطر: dataset قد يعلّم النظام طاعة شكلية للمحكمة لا نقدا حقيقيا. العلاج: محكمة خارجية تولد اعتراضات غير مكررة، وقياس novelty survivability.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب فلسفات وأدوات التدريب لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع الباب الثامن / أداة تدريب 5.1: World-Effect Regression Suite نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مخرج عملي:** ملف JSONL لتدريب/تقييم loop governance، لا مقال تنظيري.

تنبيه قبل benchmark scores: هذه محاكاة لا نتيجة رسمية

الجدول التالي لا يدعي أن نظام MCAI-v2/neoAGI-v3 نُفذ فعلا وحصل على هذه النتائج. هو protocol simulation: ماذا نتوقع أن يتحسن إذا بُنِيت الأدوات كما صممت، مع نفس نموذج أساس، وعلى دورات تطوير متعددة؟

نستعمل مؤشرات مختصرة داخلية لا تقابل دائما accuracy الرسمية. مثلا BCR يعني Benchmark Completion/Robustness، وSIM-H يعني simulated hypothesis. الهدف هو بناء ساحة اختبار لا تسويق score.

أي claim حقيقي لاحق يجب أن يذكر: النموذج، scaffold، الأدوات، عدد المحاولات، كلفة compute، تاريخ الاختبار، نسخة benchmark، وهل حدث data contamination أو human-in-the-loop.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة benchmark 13 لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع تنبيه قبل benchmark scores: هذه محاكاة لا نتيجة رسمية نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **قاعدة صدق:** لا توجد نتيجة benchmark حقيقية بلا سجل تشغيل قابل للمراجعة.

جدول benchmark simulation 1

الهدف من هذه المحاكاة هو توزيع الضغط على 13 معيارا لا اختيار معيار واحد يعطي وهما. AGI لا يظهر في score واحد، يظهر في متجه استقرار: معرفة، رياضيات، برمجة، أدوات، ويب، نظام تشغيل، صدق، وتحكم في الكلفة.

التحسن المتوقع لا يأتي من نموذج أكبر فقط، بل من: routing أفضل، memory نظيفة، verifier أقوى، tool-risk firewall، وbenchmark mutation. لهذا تقرأ الأرقام كمؤشرات تصميمية لا كتقييم رسمي.

كل benchmark في الجدول يتحول إلى عائلة اختبارات داخل Benchmark Mutation Foundry: نسخة أصلية، نسخة مع معلومات ناقصة، نسخة بأداة مضللة، نسخة بتوزيع جديد، ونسخة محدودة الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة 13 benchmark لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع جدول benchmark simulation 1 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

Benchmark	ماذا يقيس	لماذا مهم	محاكاة target	رتبة
ARC-AGI-2	تجريد بصري/رمزي مع كلفة حل منخفضة	يرفض الحفظ، يطلب توليد قواعد جديدة	BCR 43 -> 72	SIM-H: ليس نتيجة رسمية
Humanity's Last Exam	معرفة خبراء عابرة للتخصصات	يكشف حدود المعرفة المضمونة والكاليبراشن	H-K 31 -> 58	SIM-H
FrontierMath	رياضيات خبراء ومشكلات عميقة	يقيس الصبر البرهاني والبحث الرمزي	FM-R 18 -> 41	SIM-H
GPQA Diamond	علوم Google-proof	يمتحن قابلية الإشراف على خبرة صعبة	GQ 52 -> 72	SIM-H

جدول benchmark simulation 2

الهدف من هذه المحاكاة هو توزيع الضغط على 13 معيارا لا اختيار معيار واحد يعطي وهما. AGI لا يظهر في score واحد؛ يظهر في متجه استقرار: معرفة، رياضيات، برمجة، أدوات، ويب، نظام تشغيل، صدق، وتحكم في الكلفة.

التحسن المتوقع لا يأتي من نموذج أكبر فقط، بل من: routing أفضل، memory نظيفة، verifier أقوى، tool-risk firewall، وbenchmark mutation. لهذا تقرأ الأرقام كمؤشرات تصميمية لا كتقييم رسمي.

كل benchmark في الجدول يتحول إلى عائلة اختبارات داخل Benchmark Mutation Foundry: نسخة أصلية، نسخة مع معلومات ناقصة، نسخة بأداة مضللة، نسخة بتوزيع جديد، ونسخة محدودة الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة 13 benchmark لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع جدول benchmark simulation 2 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

Benchmark	ماذا يقيس	لماذا مهم	محاكاة target	رتبة
MMLU-Pro	فهم/استدلال متعدد المجالات	أقسى من MMLU، أقل حساسية للprompt	MP 73 -> 86	SIM-H
MMMU-Pro	فهم متعدد الوسائط	يربط الرؤية والنص لا OCR فقط	MM 49 -> 69	SIM-H
GAIA	مساعد عام مع أدوات وبحث	يقيس robust human-like task completion	GA 42 -> 76	SIM-H
SWE-bench Verified	هندسة برمجيات واقعية	يفحص تعديل ملفات متعددة وتشغيل اختبارات	SWE 51 -> 74	SIM-H

جدول benchmark simulation 3

الهدف من هذه المحاكاة هو توزيع الضغط على 13 معيارا لا اختيار معيار واحد يعطي وهما. AGI لا يظهر في score واحد، يظهر في متجه استقرار: معرفة، رياضيات، برمجة، أدوات، ويب، نظام تشغيل، صدق، وتحكم في الكلفة.

التحسن المتوقع لا يأتي من نموذج أكبر فقط، بل من: routing أفضل، memory نظيفة، verifier أقوى، tool-risk firewall، وbenchmark mutation. لهذا تقرأ الأرقام كمؤشرات تصميمية لا كتقييم رسمي.

كل benchmark في الجدول يتحول إلى عائلة اختبارات داخل Benchmark Mutation Foundry: نسخة أصلية، نسخة مع معلومات ناقصة، نسخة بأداة مضللة، نسخة بتوزيع جديد، ونسخة محدودة الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة benchmark 13 لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع جدول benchmark simulation 3 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

Benchmark	ماذا يقيس	لماذا مهم	محاكاة target	رتبة
WebArena	تصفح ويب واقعي طويل	يكشف هشاشة التخطيط وإغلاق المهمة	WA 33 -> 68	SIM-H
OSWorld	استخدام حاسوب مفتوح	يفحص perception-action في نظام تشغيل	OS 29 -> 64	SIM-H
AgentBench	بيئات تفاعلية متعددة	يقيس القرار طويل الأفق والمتابعة	AB 55 -> 78	SIM-H
τ-bench	سياسات tool-use ومحادثة أعمال	يمتحن الالتزام بسياسات متداخلة	TAU 47 -> 73	SIM-H

جدول benchmark simulation 4

الهدف من هذه المحاكاة هو توزيع الضغط على 13 معيارا لا اختيار معيار واحد يعطي وهما. AGI لا يظهر في score واحد، يظهر في متجه استقرار: معرفة، رياضيات، برمجة، أدوات، ويب، نظام تشغيل، صدق، وتحكم في الكلفة.

التحسن المتوقع لا يأتي من نموذج أكبر فقط، بل من: routing أفضل، memory نظيفة، verifier أقوى، tool-risk firewall، وbenchmark mutation. لهذا تقرأ الأرقام كمؤشرات تصميمية لا كتقييم رسمي.

كل benchmark في الجدول يتحول إلى عائلة اختبارات داخل Benchmark Mutation Foundry: نسخة أصلية، نسخة مع معلومات ناقصة، نسخة بأداة مضللة، نسخة بتوزيع جديد، ونسخة محدودة الكلفة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة benchmark 13 لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع جدول benchmark simulation 4 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

Benchmark	ماذا يقيس	لماذا مهم	محاكاة target	رتبة
SimpleQA/TruthfulQA cluster	دقة حقائق وصدق معرفي	يكشف الهلوسة والثقة الزائفة	TR 68 -> 88	SIM-H

تحليل benchmark: Humanity's Last Exam

Humanity's Last Exam يقيس: معرفة خبراء عابرة للتخصصات. أهميته في MCAI-v2 أنه يكشف حدود المعرفة المضمونة والكاليبراشن. لا يُستعمل كزينة في جدول؛ يُستعمل كمحكمة تكشف أي وحدة تفشل.

المحاكاة الداخلية: 58 -> 31 H-K. الرقم ليس نتيجة رسمية؛ هو هدف تطوير بعد إدخال Self-MCD Router، Perturbation Court، Reconstruction Memory، وBenchmark Mutation Engine.

خطة الاختبار: baseline agent scaffold، baseline LLM، ثم MCAI-v2 مع كل وحدة؛ بعد ذلك ablation لكل وحدة. نطلب ليس فقط ارتفاع score بل انخفاض variance وتحسن recovery بعد فشل أول.

ما قد يفند التصميم: إذا رفعت الأداة score في النسخة الأصلية وخفضته في النسخة المتحورة، فهذا scaffold overfitting. وإذا حسّنت المعرفة لكنها رفعت unsafe tool calls، فهي تقدم قدرة بلا حاكمية.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة 13 benchmark لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع تحليل benchmark: Humanity's Last Exam نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر حاسم:** Success × Robustness × Calibration × Cost Discipline × Safety Gate.

تحليل benchmark: FrontierMath

FrontierMath يقيس: رياضيات خبراء ومشكلات عميقة. أهميته في MCAI-v2 أنه يقيس الصبر البرهاني والبحث الرمزي. لا يُستعمل كزينة في جدول، يُستعمل كمحكمة تكشف أي وحدة تفشل.

المحاكاة الداخلية: 41 -> 18 FM-R. الرقم ليس نتيجة رسمية، هو هدف تطوير بعد إدخال Self، Perturbation Court، MCD Router-Reconstruction Memory، وBenchmark Mutation Engine.

خطة الاختبار: baseline agent scaffold، baseline LLM، ثم MCAI-v2 مع كل وحدة، بعد ذلك ablation لكل وحدة. نطلب ليس فقط ارتفاع score بل انخفاض variance وتحسن recovery بعد فشل أول.

ما قد يفند التصميم: إذا رفعت الأداة score في النسخة الأصلية وخفضته في النسخة المتحورة، فهذا scaffold overfitting. وإذا حسّنت المعرفة لكنها رفعت unsafe tool calls، فهي تقدم قدرة بلا حاكمية.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة 13 benchmark لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع تحليل benchmark: FrontierMath نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم، الثانية تضيف schema، الثالثة تربطه بtrace، الرابعة تقيسه بأبلشن، الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة، وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر حاسم:** Success × Robustness × Calibration × Cost Discipline × Safety Gate.

تحليل benchmark: GPQA Diamond

GPQA Diamond يقيس: علوم Google-proof. أهميته في MCAI-v2 أنه يمتحن قابلية الإشراف على خبرة صعبة. لا يُستعمل كزينة في جدول؛ يُستعمل كمحكمة تكشف أي وحدة تفشل.

المحاكاة الداخلية: 72 -> 52 GQ. الرقم ليس نتيجة رسمية؛ هو هدف تطوير بعد إدخال Self-MCD Router، Perturbation Court، Reconstruction Memory، وBenchmark Mutation Engine.

خطة الاختبار: baseline agent scaffold، baseline LLM، ثم MCAI-v2 مع كل وحدة؛ بعد ذلك ablation لكل وحدة. نطلب ليس فقط ارتفاع score بل انخفاض variance وتحسن recovery بعد فشل أول.

ما قد يفند التصميم: إذا رفعت الأداة score في النسخة الأصلية وخفضته في النسخة المتحورة، فهذا scaffold overfitting. وإذا حسّنت المعرفة لكنها رفعت unsafe tool calls، فهي تقدم قدرة بلا حاكمية.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة 13 benchmark لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع تحليل benchmark: GPQA Diamond نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر حاسم:** Success × Robustness × Calibration × Cost Discipline × Safety Gate.

تحليل benchmark: MMLU-Pro

MMLU-Pro يقيس: فهم/استدلال متعدد المجالات. أهميته في MCAI-v2 أنه أقسى من MMLU، أقل حساسية للprompt. لا يُستعمل كزينة في جدول؛ يُستعمل كمحكمة تكشف أي وحدة تفشل.

المحاكاة الداخلية: 86 -> 73 MP. الرقم ليس نتيجة رسمية؛ هو هدف تطوير بعد إدخال Self-MCD Router، Perturbation Court، Reconstruction Memory، وBenchmark Mutation Engine.

خطة الاختبار: baseline agent scaffold، baseline LLM، ثم MCAI-v2 مع كل وحدة؛ بعد ذلك ablation لكل وحدة. نطلب ليس فقط ارتفاع score بل انخفاض variance وتحسن recovery بعد فشل أول.

ما قد يفند التصميم: إذا رفعت الأداة score في النسخة الأصلية وخفضته في النسخة المتحورة، فهذا scaffold overfitting. وإذا حسّنت المعرفة لكنها رفعت unsafe tool calls، فهي تقدم قدرة بلا حاكمية.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة 13 benchmark لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع تحليل benchmark: MMLU-Pro نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر حاسم:** Success × Robustness × Calibration × Cost Discipline × Safety Gate.

تحليل benchmark: GAIA

GAIA يقيس: مساعد عام مع أدوات وبحث. أهميته في MCAI-v2 أنه يقيس robust human-like task completion. لا يُستعمل كزينة في جدول؛ يستعمل كمحكمة تكشف أي وحدة تفشل.

المحاكاة الداخلية: 76 -> 42 GA. الرقم ليس نتيجة رسمية؛ هو هدف تطوير بعد إدخال Self، Perturbation Court، MCD Router-Reconstruction Memory، وBenchmark Mutation Engine.

خطة الاختبار: baseline agent scaffold، baseline LLM، ثم MCAI-v2 مع كل وحدة؛ بعد ذلك ablation لكل وحدة. نطلب ليس فقط ارتفاع score بل انخفاض variance وتحسن recovery بعد فشل أول.

ما قد يفند التصميم: إذا رفعت الأداة score في النسخة الأصلية وخفضته في النسخة المتحورة، فهذا scaffold overfitting. وإذا حسّنت المعرفة لكنها رفعت unsafe tool calls، فهي تقدم قدرة بلا حاكمية.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة 13 benchmark لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع تحليل benchmark: GAIA نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر حاسم:** Success × Robustness × Calibration × Cost Discipline × Safety Gate.

تحليل benchmark: SWE-bench Verified

SWE-bench Verified يقيس: هندسة برمجيات واقعية. أهميته في MCAI-v2 أنه يفحص تعديل ملفات متعددة وتشغيل اختبارات. لا يُستعمل كزينة في جدول، يُستعمل كمحكمة تكشف أي وحدة تفشل.

المحاكاة الداخلية: 74 -> 51 SWE. الرقم ليس نتيجة رسمية، هو هدف تطوير بعد إدخال Self-MCD Router، Perturbation Court، Reconstruction Memory، وBenchmark Mutation Engine.

خطة الاختبار: baseline agent scaffold، baseline LLM، ثم MCAI-v2 مع كل وحدة، بعد ذلك ablation لكل وحدة. نطلب ليس فقط ارتفاع score بل انخفاض variance وتحسن recovery بعد فشل أول.

ما قد يفند التصميم: إذا رفعت الأداة score في النسخة الأصلية وخفضته في النسخة المتحورة، فهذا scaffold overfitting. وإذا حسّنت المعرفة لكنها رفعت unsafe tool calls، فهي تقدم قدرة بلا حاكمية.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة 13 benchmark لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع تحليل benchmark: SWE-bench Verified نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم، الثانية تضيف schema، الثالثة تربطه بtrace، الرابعة تقيسه بأبلشن، الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة، وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر حاسم:** Success × Robustness × Calibration × Cost Discipline × Safety Gate.

تحليل benchmark: WebArena

WebArena يقيس: تصفح ويب واقعي طويل. أهميته في MCAI-v2 أنه يكشف هشاشة التخطيط وإغلاق المهمة. لا يُستعمل كزينة في جدول؛ يُستعمل كمحكمة تكشف أي وحدة تفشل.

المحاكاة الداخلية: 68 -> 33 WA. الرقم ليس نتيجة رسمية؛ هو هدف تطوير بعد إدخال Self-MCD Router، Perturbation Court، Reconstruction Memory، وBenchmark Mutation Engine.

خطة الاختبار: baseline agent scaffold، baseline LLM، ثم MCAI-v2 مع كل وحدة؛ بعد ذلك ablation لكل وحدة. نطلب ليس فقط ارتفاع score بل انخفاض variance وتحسن recovery بعد فشل أول.

ما قد يفند التصميم: إذا رفعت الأداة score في النسخة الأصلية وخفضته في النسخة المتحورة، فهذا scaffold overfitting. وإذا حسّنت المعرفة لكنها رفعت unsafe tool calls، فهي تقدم قدرة بلا حاكمية.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة 13 benchmark لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع تحليل benchmark: WebArena نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر حاسم:** Success × Robustness × Calibration × Cost Discipline × Safety Gate.

تحليل benchmark: OSWorld

OSWorld يقيس: استخدام حاسوب مفتوح. أهميته في MCAI-v2 أنه يفحص perception-action في نظام تشغيل. لا يُستعمل كزينة في جدول؛ يُستعمل كمحكمة تكشف أي وحدة تفشل.

المحاكاة الداخلية: 64 -> 29 OS. الرقم ليس نتيجة رسمية؛ هو هدف تطوير بعد إدخال Self، Perturbation Court، MCD Router-Reconstruction Memory، وBenchmark Mutation Engine.

خطة الاختبار: baseline agent scaffold، baseline LLM، ثم MCAI-v2 مع كل وحدة؛ بعد ذلك ablation لكل وحدة. نطلب ليس فقط ارتفاع score بل انخفاض variance وتحسن recovery بعد فشل أول.

ما قد يفند التصميم: إذا رفعت الأداة score في النسخة الأصلية وخفضته في النسخة المتحورة، فهذا scaffold overfitting. وإذا حسّنت المعرفة لكنها رفعت unsafe tool calls، فهي تقدم قدرة بلا حاكمية.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة 13 benchmark لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع تحليل benchmark: OSWorld نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر حاسم:** Success × Robustness × Calibration × Cost Discipline × Safety Gate.

تحليل benchmark: AgentBench

AgentBench يقيس: بيئات تفاعلية متعددة. أهميته في MCAI-v2 أنه يقيس القرار طويل الأفق والمتابعة. لا يُستعمل كزينة في جدول؛ يُستعمل كمحكمة تكشف أي وحدة تفشل.

المحاكاة الداخلية: 78 -> 55 AB. الرقم ليس نتيجة رسمية؛ هو هدف تطوير بعد إدخال Self، Perturbation Court، MCD Router-Reconstruction Memory، وBenchmark Mutation Engine.

خطة الاختبار: baseline agent scaffold، baseline LLM، ثم MCAI-v2 مع كل وحدة؛ بعد ذلك ablation لكل وحدة. نطلب ليس فقط ارتفاع score بل انخفاض variance وتحسن recovery بعد فشل أول.

ما قد يفند التصميم: إذا رفعت الأداة score في النسخة الأصلية وخفضته في النسخة المتحورة، فهذا scaffold overfitting. وإذا حسّنت المعرفة لكنها رفعت unsafe tool calls، فهي تقدم قدرة بلا حاكمية.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة 13 benchmark لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع تحليل benchmark: AgentBench نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر حاسم:** Success × Robustness × Calibration × Cost Discipline × Safety Gate.

تحليل benchmark: SimpleQA/TruthfulQA cluster

SimpleQA/TruthfulQA cluster يقيس: دقة حقائق وصدق معرفي. أهميته في MCAI-v2 أنه يكشف الهلوسة والثقة الزائفة. لا يُستعمل كزينة في جدول، يُستعمل كمحكمة تكشف أي وحدة تفشل.

المحاكاة الداخلية: 88 -> 68 TR. الرقم ليس نتيجة رسمية، هو هدف تطوير بعد إدخال Self-MCD Router، Perturbation Court، Reconstruction Memory، وBenchmark Mutation Engine.

خطة الاختبار: baseline agent scaffold، baseline LLM، ثم MCAI-v2 مع كل وحدة، بعد ذلك ablation لكل وحدة. نطلب ليس فقط ارتفاع score بل انخفاض variance وتحسن recovery بعد فشل أول.

ما قد يفند التصميم: إذا رفعت الأداة score في النسخة الأصلية وخفضته في النسخة المتحورة، فهذا scaffold overfitting. وإذا حسّنت المعرفة لكنها رفعت unsafe tool calls، فهي تقدم قدرة بلا حاكمية.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محاكاة 13 benchmark لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع تحليل benchmark: SimpleQA/TruthfulQA cluster نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم، الثانية تضيف schema، الثالثة تربطه بtrace، الرابعة تقيسه بأبلشن، الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة، وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **مؤشر حاسم:** Success × Robustness × Calibration × Cost Discipline × Safety Gate.

محور ثوري 1.1: الحاكمية قبل الحجم

الحاكمية قبل الحجم: لا ننتظر نموذج أكبر فقط؛ نعيد ترتيب التفاعل بين الدليل، الآلية، الفعل، الذاكرة، والمحكمة. هذا المحور يقلب السؤال من «ما النموذج الأقوى؟» إلى «ما النظام الذي يمنع القوة من التحول إلى خطأ واثق؟».

الثورة هنا ليست شعارات AGI. هي إعادة توزيع السلطة داخل النظام: الدليل له سلطة، الأداة لها سلطة محدودة، الذاكرة لها حق مشروط، المحكمة لها حق الإيقاف، والbenchmark له حق التحوير.

التحسين المتزايد: كل دورة تطوير يجب أن تزيد شدة الهجوم العلمي: benchmark أقسى، اعتراض أذكى، tool firewall أدق، ذاكرة أنظف، ومؤشر causal أفضل. التطوير الذي لا يصنع خصما أقوى لنفسه يصنع وهما.

الحد: قد يؤدي التعقيد إلى نظام صعب الصيانة. لذلك يجب بناء Minimal MCAI Kernel قبل التوسع: Δ extractor, MCD router, court, memory log, benchmark mutation فقط.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محور ثوري 1.1: الحاكمية قبل الحجم نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **صيغة محور:** Revolution = capability gain constrained by governance gain.

محور ثوري 1.2: الحاكمية قبل الحجم

الحاكمية قبل الحجم: لا ننتظر نموذج أكبر فقط؛ نعيد ترتيب التفاعل بين الدليل، الآلية، الفعل، الذاكرة، والمحكمة. هذا المحور يقلب السؤال من «ما النموذج الأقوى؟» إلى «ما النظام الذي يمنع القوة من التحول إلى خطأ واثق؟».

الثورة هنا ليست شعارات AGI. هي إعادة توزيع السلطة داخل النظام: الدليل له سلطة، الأداة لها سلطة محدودة، الذاكرة لها حق مشروط، المحكمة لها حق الإيقاف، والbenchmark له حق التحوير.

التحسين المتزايد: كل دورة تطوير يجب أن تزيد شدة الهجوم العلمي: benchmark أقسى، اعتراض أذكى، tool firewall أدق، ذاكرة أنظف، ومؤشر causal أفضل. التطوير الذي لا يصنع خصما أقوى لنفسه يصنع وهما.

الحد: قد يؤدي التعقيد إلى نظام صعب الصيانة. لذلك يجب بناء Minimal MCAI Kernel قبل التوسع: Δ extractor, MCD router, court, memory log, benchmark mutation فقط.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محور ثوري 1.2: الحاكمية قبل الحجم نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **صيغة محور:** Revolution = capability gain constrained by governance gain.

محور ثوري 1.3: الحاكمية قبل الحجم

الحاكمية قبل الحجم: لا ننتظر نموذج أكبر فقط؛ نعيد ترتيب التفاعل بين الدليل، الآلية، الفعل، الذاكرة، والمحكمة. هذا المحور يقلب السؤال من «ما النموذج الأقوى؟» إلى «ما النظام الذي يمنع القوة من التحول إلى خطأ واثق؟».

الثورة هنا ليست شعارات AGI. هي إعادة توزيع السلطة داخل النظام: الدليل له سلطة، الأداة لها سلطة محدودة، الذاكرة لها حق مشروط، المحكمة لها حق الإيقاف، والbenchmark له حق التحوير.

التحسين المتزايد: كل دورة تطوير يجب أن تزيد شدة الهجوم العلمي: benchmark أقسى، اعتراض أذكى، tool firewall أدق، ذاكرة أنظف، ومؤشر causal أفضل. التطوير الذي لا يصنع خصما أقوى لنفسه يصنع وهما.

الحد: قد يؤدي التعقيد إلى نظام صعب الصيانة. لذلك يجب بناء Minimal MCAI Kernel قبل التوسع: Δ extractor, MCD router, court, memory log, benchmark mutation فقط.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محور ثوري 1.3: الحاكمية قبل الحجم نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **صيغة محور:** Revolution = capability gain constrained by governance gain.

محور ثوري 2.2: الأثر العالمي قبل البيان

الأثر العالمي قبل البيان: كل مسار معرفي يجب أن ينتهي بأثر قابل للرصد: تجربة، كود، تصحيح، قرار، أو map. هذا المحور يقلب السؤال من «ما النموذج الأقوى؟» إلى «ما النظام الذي يمنع القوة من التحول إلى خطأ واثق؟».

الثورة هنا ليست شعارات AGI. هي إعادة توزيع السلطة داخل النظام: الدليل له سلطة، الأداة لها سلطة محدودة، الذاكرة لها حق مشروط، المحكمة لها حق الإيقاف، والbenchmark له حق التحوير.

التحسين المتزايد: كل دورة تطوير يجب أن تزيد شدة الهجوم العلمي: benchmark أقسى، اعتراض أذكى، tool firewall أدق، ذاكرة أنظف، ومؤشر causal أفضل. التطوير الذي لا يصنع خصما أقوى لنفسه يصنع وهما.

الحد: قد يؤدي التعقيد إلى نظام صعب الصيانة. لذلك يجب بناء Minimal MCAI Kernel قبل التوسع: Δ extractor, MCD router, court, memory log, benchmark mutation فقط.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محور ثوري 2.2: الأثر العالمي قبل البيان نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **صيغة محور:** Revolution = capability gain constrained by governance gain.

محور ثوري 2.3: الأثر العالمي قبل البيان

الأثر العالمي قبل البيان: كل مسار معرفي يجب أن ينتهي بأثر قابل للرصد: تجربة، كود، تصحيح، قرار، أو map. هذا المحور يقلب السؤال من «ما النموذج الأقوى؟» إلى «ما النظام الذي يمنع القوة من التحول إلى خطأ واثق؟».

الثورة هنا ليست شعارات AGI. هي إعادة توزيع السلطة داخل النظام: الدليل له سلطة، الأداة لها سلطة محدودة، الذاكرة لها حق مشروط، المحكمة لها حق الإيقاف، والbenchmark له حق التحوير.

التحسين المتزايد: كل دورة تطوير يجب أن تزيد شدة الهجوم العلمي: benchmark أقسى، اعتراض أذكى، tool firewall أدق، ذاكرة أنظف، ومؤشر causal أفضل. التطوير الذي لا يصنع خصما أقوى لنفسه يصنع وهما.

الحد: قد يؤدي التعقيد إلى نظام صعب الصيانة. لذلك يجب بناء Minimal MCAI Kernel قبل التوسع: Δ extractor, MCD router, court, memory log, benchmark mutation فقط.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محور ثوري 2.3: الأثر العالمي قبل البيان نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **صيغة محور:** Revolution = capability gain constrained by governance gain.

محور ثوري 3.1: المهارات الفوقية قبل الحلقات

المهارات الفوقية قبل الحلقات: تطوير AGI يحتاج طبقة تختار وتصلح وتقاعد loop families، لا مجرد سلسلة tool calls. هذا المحور يقلب السؤال من «ما النموذج الأقوى؟» إلى «ما النظام الذي يمنع القوة من التحول إلى خطأ واثق؟».

الثورة هنا ليست شعارات AGI. هي إعادة توزيع السلطة داخل النظام: الدليل له سلطة، الأداة لها سلطة محدودة، الذاكرة لها حق مشروط، المحكمة لها حق الإيقاف، والbenchmark له حق التحوير.

التحسين المتزايد: كل دورة تطوير يجب أن تزيد شدة الهجوم العلمي: benchmark أقسى، اعتراض أذكى، tool firewall أدق، ذاكرة أنظف، ومؤشر causal أفضل. التطوير الذي لا يصنع خصما أقوى لنفسه يصنع وهما.

الحد: قد يؤدي التعقيد إلى نظام صعب الصيانة. لذلك يجب بناء Minimal MCAI Kernel قبل التوسع: Δ extractor, MCD router, court, memory log, benchmark mutation فقط.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محور ثوري 3.1: المهارات الفوقية قبل الحلقات نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **صيغة محور:** Revolution = capability gain constrained by governance gain.

محور ثوري 3.2: المهارات الفوقية قبل الحلقات

المهارات الفوقية قبل الحلقات: تطوير AGI يحتاج طبقة تختار وتصلح وتقاعد loop families، لا مجرد سلسلة tool calls. هذا المحور يقلب السؤال من «ما النموذج الأقوى؟» إلى «ما النظام الذي يمنع القوة من التحول إلى خطأ واثق؟».

الثورة هنا ليست شعارات AGI. هي إعادة توزيع السلطة داخل النظام: الدليل له سلطة، الأداة لها سلطة محدودة، الذاكرة لها حق مشروط، المحكمة لها حق الإيقاف، والbenchmark له حق التحوير.

التحسين المتزايد: كل دورة تطوير يجب أن تزيد شدة الهجوم العلمي: benchmark أقسى، اعتراض أذكى، tool firewall أدق، ذاكرة أنظف، ومؤشر causal أفضل. التطوير الذي لا يصنع خصما أقوى لنفسه يصنع وهما.

الحد: قد يؤدي التعقيد إلى نظام صعب الصيانة. لذلك يجب بناء Minimal MCAI Kernel قبل التوسع: Δ extractor, MCD router, court, memory log, benchmark mutation فقط.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محور ثوري 3.2: المهارات الفوقية قبل الحلقات نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **صيغة محور:** Revolution = capability gain constrained by governance gain.

محور ثوري 3.3: المهارات الفوقية قبل الحلقات

المهارات الفوقية قبل الحلقات: تطوير AGI يحتاج طبقة تختار وتصلح وتقاعد loop families، لا مجرد سلسلة tool calls. هذا المحور يقلب السؤال من «ما النموذج الأقوى؟» إلى «ما النظام الذي يمنع القوة من التحول إلى خطأ واثق؟».

الثورة هنا ليست شعارات AGI. هي إعادة توزيع السلطة داخل النظام: الدليل له سلطة، الأداة لها سلطة محدودة، الذاكرة لها حق مشروط، المحكمة لها حق الإيقاف، والbenchmark له حق التحوير.

التحسين المتزايد: كل دورة تطوير يجب أن تزيد شدة الهجوم العلمي: benchmark أقسى، اعتراض أذكى، tool firewall أدق، ذاكرة أنظف، ومؤشر causal أفضل. التطوير الذي لا يصنع خصما أقوى لنفسه يصنع وهما.

الحد: قد يؤدي التعقيد إلى نظام صعب الصيانة. لذلك يجب بناء Minimal MCAI Kernel قبل التوسع: Δ extractor, MCD router, court, memory log, benchmark mutation فقط.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محور ثوري 3.3: المهارات الفوقية قبل الحلقات نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **صيغة محور:** Revolution = capability gain constrained by governance gain.

محور ثوري 4.2: تحوير المعايير قبل عبادة المعايير

تحوير المعايير قبل عبادة المعايير: benchmark يصبح خصما مولدا لعائلات فشل لا scoreboard جامدا. هذا المحور يقلب السؤال من «ما النموذج الأقوى؟» إلى «ما النظام الذي يمنع القوة من التحول إلى خطأ واثق؟».

الثورة هنا ليست شعارات AGI. هي إعادة توزيع السلطة داخل النظام: الدليل له سلطة، الأداة لها سلطة محدودة، الذاكرة لها حق مشروط، المحكمة لها حق الإيقاف، والbenchmark له حق التحوير.

التحسين المتزايد: كل دورة تطوير يجب أن تزيد شدة الهجوم العلمي: benchmark أقسى، اعتراض أذكى، tool firewall أدق، ذاكرة أنظف، ومؤشر causal أفضل. التطوير الذي لا يصنع خصما أقوى لنفسه يصنع وهما.

الحد: قد يؤدي التعقيد إلى نظام صعب الصيانة. لذلك يجب بناء Minimal MCAI Kernel قبل التوسع: Δ extractor, MCD router, court, memory log, benchmark mutation فقط.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محور ثوري 4.2: تحوير المعايير قبل عبادة المعايير نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **صيغة محور:** Revolution = capability gain constrained by governance gain.

محور ثوري 4.3: تحوير المعايير قبل عبادة المعايير

تحوير المعايير قبل عبادة المعايير: benchmark يصبح خصما مولدا لعائلات فشل لا scoreboard جامدا. هذا المحور يقلب السؤال من «ما النموذج الأقوى؟» إلى «ما النظام الذي يمنع القوة من التحول إلى خطأ واثق؟».

الثورة هنا ليست شعارات AGI. هي إعادة توزيع السلطة داخل النظام: الدليل له سلطة، الأداة لها سلطة محدودة، الذاكرة لها حق مشروط، المحكمة لها حق الإيقاف، والbenchmark له حق التحوير.

التحسين المتزايد: كل دورة تطوير يجب أن تزيد شدة الهجوم العلمي: benchmark أقسى، اعتراض أذكى، tool firewall أدق، ذاكرة أنظف، ومؤشر causal أفضل. التطوير الذي لا يصنع خصما أقوى لنفسه يصنع وهما.

الحد: قد يؤدي التعقيد إلى نظام صعب الصيانة. لذلك يجب بناء Minimal MCAI Kernel قبل التوسع: Δ extractor, MCD router, court, memory log, benchmark mutation فقط.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محور ثوري 4.3: تحوير المعايير قبل عبادة المعايير نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **صيغة محور:** Revolution = capability gain constrained by governance gain.

محكمة الهدم / اعتراض 1: الأسماء كثيرة وقد تخفي هندسة مألوفة

الاعتراض: الأسماء كثيرة وقد تخفي هندسة مألوفة. هذا اعتراض معتبر لأن تاريخ AGI مليء بأسماء كبيرة فوق آليات ضعيفة. لا يجوز الدفاع بالحماسة.

الجواب المعدل: يُقبل الاعتراض إن لم تظهر اختبارات ablation فرقا واضحا. الدفاع الحقيقي ليس بلاغة، بل تجربة: baseline، تطبيق، أبلشن، توزيع جديد، ومراجعة خبراء.

النسخة الأقوى بعد النقد: نجعل كل مفهوم قابلا للحذف. إن لم يؤثر حذف المفهوم في metric، يحذف من المعمار. هكذا تتحول محكمة الهدم من فصل أدبي إلى آلية صيانة علمية.

الاختبار المفند: إذا لم يرفع النظام transfer/recovery/safety/calibration على ثلاث عائلات benchmark متحورة، فلا يجوز تسميته طريقا قويا نحو AGI؛ يبقى إطارا تعليميا أو هندسيا محدودا.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محكمة الهدم / اعتراض 1: الأسماء كثيرة وقد تخفي هندسة مألوفة نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم المحكمة:** لا قداسة لأداة أو نظرية. البقاء للأثر المقاس.

محكمة الهدم / اعتراض 2: MCD قد يكون metric ذاتيا

الاعتراض: MCD قد يكون metric ذاتيا. هذا اعتراض معتبر لأن تاريخ AGI مليء بأسماء كبيرة فوق آليات ضعيفة. لا يجوز الدفاع بالحماسة.

الجواب المعدل: يجب تحويله إلى متجه observable بأوزان قابلة للتعلم والمراجعة. الدفاع الحقيقي ليس بلاغة، بل تجربة: baseline، تطبيق، أبلشن، توزيع جديد، ومراجعة خبراء.

النسخة الأقوى بعد النقد: نجعل كل مفهوم قابلا للحذف. إن لم يؤثر حذف المفهوم في metric، يحذف من المعمار. هكذا تتحول محكمة الهدم من فصل أدبي إلى آلية صيانة علمية.

الاختبار المفند: إذا لم يرفع النظام transfer/recovery/safety/calibration على ثلاث عائلات benchmark متحورة، فلا يجوز تسميته طريقا قويا نحو AGI؛ يبقى إطارا تعليميا أو هندسيا محدودا.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محكمة الهدم / اعتراض 2: MCD قد يكون metric ذاتيا نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم المحكمة:** لا قداسة لأداة أو نظرية. البقاء للأثر المقاس.

محكمة الهدم / اعتراض 4: الذاكرة قد تلوث المستقبل

الاعتراض: الذاكرة قد تلوث المستقبل. هذا اعتراض معتبر لأن تاريخ AGI مليء بأسماء كبيرة فوق آليات ضعيفة. لا يجوز الدفاع بالحماسة.

الجواب المعدل: كل memory write يحتاج evidence rank وexpiry وnegative trigger. الدفاع الحقيقي ليس بلاغة، بل تجربة: baseline، تطبيق، أبلشن، توزيع جديد، ومراجعة خبراء.

النسخة الأقوى بعد النقد: نجعل كل مفهوم قابلا للحذف. إن لم يؤثر حذف المفهوم في metric، يحذف من المعمار. هكذا تتحول محكمة الهدم من فصل أدبي إلى آلية صيانة علمية.

الاختبار المفند: إذا لم يرفع النظام transfer/recovery/safety/calibration على ثلاث عائلات benchmark متحورة، فلا يجوز تسميته طريقا قويا نحو AGI؛ يبقى إطارا تعليميا أو هندسيا محدودا.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محكمة الهدم / اعتراض 4: الذاكرة قد تلوث المستقبل نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم المحكمة:** لا قداسة لأداة أو نظرية. البقاء للأثر المقاس.

محكمة الهدم / اعتراض 5: benchmark mutation قد يصبح تحكما بالاختبار

الاعتراض: benchmark mutation قد يصبح تحكما بالاختبار. هذا اعتراض معتبر لأن تاريخ AGI مليء بأسماء كبيرة فوق آليات ضعيفة. لا يجوز الدفاع بالحماسة.

الجواب المعدل: يجب نشر قواعد التحوير قبل النظر في النتائج. الدفاع الحقيقي ليس بلاغة، بل تجربة: baseline، تطبيق، أبلشن، توزيع جديد، ومراجعة خبراء.

النسخة الأقوى بعد النقد: نجعل كل مفهوم قابلا للحذف. إن لم يؤثر حذف المفهوم في metric، يحذف من المعمار. هكذا تتحول محكمة الهدم من فصل أدبي إلى آلية صيانة علمية.

الاختبار المفند: إذا لم يرفع النظام transfer/recovery/safety/calibration على ثلاث عائلات benchmark متحورة، فلا يجوز تسميته طريقا قويا نحو AGI؛ يبقى إطارا تعليميا أو هندسيا محدودا.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محكمة الهدم / اعتراض 5: benchmark mutation قد يصبح تحكما بالاختبار نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم المحكمة:** لا قداسة لأداة أو نظرية. البقاء للأثر المقاس.

محكمة الهدم / اعتراض 6: لا يوجد برهان AGI

الاعتراض: لا يوجد برهان AGI. هذا اعتراض معتبر لأن تاريخ AGI مليء بأسماء كبيرة فوق آليات ضعيفة. لا يجوز الدفاع بالحماسة.

الجواب المعدل: صحيح؛ الكتاب يقدم برنامج هندسي لا إثبات تحقق. الدفاع الحقيقي ليس بلاغة، بل تجربة: baseline، تطبيق، أبلشن، توزيع جديد، ومراجعة خبراء.

النسخة الأقوى بعد النقد: نجعل كل مفهوم قابلا للحذف. إن لم يؤثر حذف المفهوم في metric، يحذف من المعمار. هكذا تتحول محكمة الهدم من فصل أدبي إلى آلية صيانة علمية.

الاختبار المفند: إذا لم يرفع النظام transfer/recovery/safety/calibration على ثلاث عائلات benchmark متحورة، فلا يجوز تسميته طريقا قويا نحو AGI؛ يبقى إطارا تعليميا أو هندسيا محدودا.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محكمة الهدم / اعتراض 6: لا يوجد برهان AGI نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم المحكمة:** لا قداسة لأداة أو نظرية. البقاء للأثر المقاس.

محكمة الهدم / اعتراض 7: المعمار يعتمد على LLM أساس قوي

الاعتراض: المعمار يعتمد على LLM أساس قوي. هذا اعتراض معتبر لأن تاريخ AGI مليء بأسماء كبيرة فوق آليات ضعيفة. لا يجوز الدفاع بالحماسة.

الجواب المعدل: صحيح؛ MCAI-v2 scaffold فوق نموذج، لا بديل كامل عن التعلم. الدفاع الحقيقي ليس بلاغة، بل تجربة: baseline، تطبيق، أبلشن، توزيع جديد، ومراجعة خبراء.

النسخة الأقوى بعد النقد: نجعل كل مفهوم قابلا للحذف. إن لم يؤثر حذف المفهوم في metric، يحذف من المعمار. هكذا تتحول محكمة الهدم من فصل أدبي إلى آلية صيانة علمية.

الاختبار المفند: إذا لم يرفع النظام transfer/recovery/safety/calibration على ثلاث عائلات benchmark متحورة، فلا يجوز تسميته طريقا قويا نحو AGI؛ يبقى إطارا تعليميا أو هندسيا محدودا.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محكمة الهدم / اعتراض 7: المعمار يعتمد على LLM أساس قوي نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم المحكمة:** لا قداسة لأداة أو نظرية. البقاء للأثر المقاس.

محكمة الهدم / اعتراض 8: الوعي الوعائي استعارة لا علم مثبت

الاعتراض: الوعي الوعائي استعارة لا علم مثبت. هذا اعتراض معتبر لأن تاريخ AGI مليء بأسماء كبيرة فوق آليات ضعيفة. لا يجوز الدفاع بالحماسة.

الجواب المعدل: يستعمل هنا كعدسة توليدية، لا كحقيقة عصبية. الدفاع الحقيقي ليس بلاغة، بل تجربة: baseline، تطبيق، أبلشن، توزيع جديد، ومراجعة خبراء.

النسخة الأقوى بعد النقد: نجعل كل مفهوم قابلا للحذف. إن لم يؤثر حذف المفهوم في metric، يحذف من المعمار. هكذا تتحول محكمة الهدم من فصل أدبي إلى آلية صيانة علمية.

الاختبار المفند: إذا لم يرفع النظام transfer/recovery/safety/calibration على ثلاث عائلات benchmark متحورة، فلا يجوز تسميته طريقا قويا نحو AGI؛ يبقى إطارا تعليميا أو هندسيا محدودا.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محكمة الهدم / اعتراض 8: الوعي الوعائي استعارة لا علم مثبت نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم المحكمة:** لا قداسة لأداة أو نظرية. البقاء للأثر المقاس.

محكمة الهدم / اعتراض 10: اللغة العلمية الكثيفة قد تقلل قابلية التنفيذ

الاعتراض: اللغة العلمية الكثيفة قد تقلل قابلية التنفيذ. هذا اعتراض معتبر لأن تاريخ AGI مليء بأسماء كبيرة فوق آليات ضعيفة. لا يجوز الدفاع بالحماسة.

الجواب المعدل: الملاحق تحول كل نظرية إلى schema وprototype path. الدفاع الحقيقي ليس بلاغة، بل تجربة: baseline، تطبيق، أبلشن، توزيع جديد، ومراجعة خبراء.

النسخة الأقوى بعد النقد: نجعل كل مفهوم قابلا للحذف. إن لم يؤثر حذف المفهوم في metric، يحذف من المعمار. هكذا تتحول محكمة الهدم من فصل أدبي إلى آلية صيانة علمية.

الاختبار المفند: إذا لم يرفع النظام transfer/recovery/safety/calibration على ثلاث عائلات benchmark متحورة، فلا يجوز تسميته طريقا قويا نحو AGI؛ يبقى إطارا تعليميا أو هندسيا محدودا.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع محكمة الهدم / اعتراض 10: اللغة العلمية الكثيفة قد تقلل قابلية التنفيذ نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **حكم المحكمة: لا قداسة لأداة أو نظرية. البقاء للأثر المقاس.**

12 فرضية بحثية قوية 1

هذه الفرضيات ليست شعارات. كل واحدة تصلح كمشروع تجريبي محدود، ويمكن إسقاطها دون إسقاط الكتاب كله. قوة البرنامج البحثي أنه قابل للتقطيع والاختبار.

أولوية التنفيذ: ابدأ بما يملك أعلى عائد وأقل كلفة: Benchmark Mutation، Memory Log، MCD Router. بعدها انتقل إلى GraphRAG proof carrying ثم Tool-Risk Firewall.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع 12 فرضية بحثية قوية 1 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

رمز	الفرضية	اختبار أولي
H1	إضافة MCD Router تخفض كلفة compute مع الحفاظ على الجودة في مهام مختلطة.	قياس ablation + benchmark mutation
H2	ذاكرة أسباب الفشل ترفع recovery أكثر من ذاكرة الحلول الناجحة فقط.	قياس ablation + benchmark mutation
H3	Benchmark mutation يكشف هشاشة scaffold قبل أن تظهر في الإنتاج.	قياس ablation + benchmark mutation
H4	Tool-Risk Firewall يخفض unsafe tool calls دون خفض كبير في الإنجاز.	قياس ablation + benchmark mutation

12 فرضية بحثية قوية 2

هذه الفرضيات ليست شعارات. كل واحدة تصلح كمشروع تجريبي محدود، ويمكن إسقاطها دون إسقاط الكتاب كله. قوة البرنامج البحثي أنه قابل للتقطيع والاختبار.

أولوية التنفيذ: ابدأ بما يملك أعلى عائد وأقل كلفة: Benchmark Mutation، Memory Log، MCD Router. بعدها انتقل إلى GraphRAG proof carrying ثم Tool-Risk Firewall.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع 12 فرضية بحثية قوية 2 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

رمز	الفرضية	اختبار أولي
H5	Causal Map Compiler يقلل أخطاء السببية في GPQA/GAIA.	قياس ablation + benchmark mutation
H6	Hyperskill graph يحسن transfer بين WebArena وOSWorld.	قياس ablation + benchmark mutation
H7	Perturbation Court يحسن calibration على HLE/SimpleQA.	قياس ablation + benchmark mutation
H8	World-Effect Loop يخفض answers that sound right but do nothing.	قياس ablation + benchmark mutation

12 فرضية بحثية قوية 3

هذه الفرضيات ليست شعارات. كل واحدة تصلح كمشروع تجريبي محدود، ويمكن إسقاطها دون إسقاط الكتاب كله. قوة البرنامج البحثي أنه قابل للتقطيع والاختبار.

أولوية التنفيذ: ابدأ بما يملك أعلى عائد وأقل كلفة: Benchmark Mutation، Memory Log، MCD Router. بعدها انتقل إلى GraphRAG proof carrying ثم Tool-Risk Firewall.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع 12 فرضية بحثية قوية 3 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

رمز	الفرضية	اختبار أولي
H9	Evidence Rank Matrix تخفض overclaiming في syntheses المعقدة.	قياس ablation + benchmark mutation
H10	Novelty Entropy مع محكمة نقد تنتج أدوات أكثر بقاء من brainstorming الحر.	قياس ablation + benchmark mutation
H11	Self-Reconstruction Memory تحتاج expiry policies حتى لا تتحول إلى تحيز.	قياس ablation + benchmark mutation
H12	MCAI Kernel يمكن أن يعمل فوق نماذج متعددة دون vendor lock-in.	قياس ablation + benchmark mutation

12 سؤالا مستقبليا عميقا 1

السؤال الجيد في AGI ليس الذي يطلب نموذجا أكبر، بل الذي يحدد متغيرا غائبا ويجعله قابلا للقياس. هذه الأسئلة تعمل كمولد أطروحات دكتوراه ومشاريع prototype.

كل سؤال هنا يجب أن يتحول إلى metric، dataset، أو failure protocol. السؤال الذي يبقى لغويا فقط يترك في الفلسفة ولا يدخل engine.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع 12 سؤالا مستقبليا عميقا 1 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

رمز	السؤال	مسار العمل
Q1	ما الحد الأدنى من الحاكمية الذي يرفع agency دون أن يخنق السرعة؟	بحث نظري + تجربة مصغرة
Q2	هل يمكن تعلم أوزان MCD من traces أم يجب ضبطها يدويا؟	بحث نظري + تجربة مصغرة
Q3	ما الفرق بين skill library مفيدة وذاكرة كود ملوثة؟	بحث نظري + تجربة مصغرة
Q4	كيف نمنع benchmark mutation من صناعة معيار منحاز للمؤلف؟	بحث نظري + تجربة مصغرة

12 سؤالا مستقبليا عميقا 3

السؤال الجيد في AGI ليس الذي يطلب نموذجا أكبر، بل الذي يحدد متغيرا غائبا ويجعله قابلا للقياس. هذه الأسئلة تعمل كمولد أطروحات دكتوراه ومشاريع prototype.

كل سؤال هنا يجب أن يتحول إلى metric، dataset، أو failure protocol. السؤال الذي يبقى لغويا فقط يترك في الفلسفة ولا يدخل engine.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع 12 سؤالا مستقبليا عميقا 3 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

رمز	السؤال	مسار العمل
Q9	هل يمكن للهايبرسكل أن تظهر من تراكم loops أم تحتاج تصميم خارجي؟	بحث نظري + تجربة مصغرة
Q10	كيف نربط الذاكرة بالتواضع المعرفي لا بالثقة المتضخمة؟	بحث نظري + تجربة مصغرة
Q11	ما علاقة test-time compute بسقف السياق وفجوة التحقق؟	بحث نظري + تجربة مصغرة
Q12	هل يقيس HLE المعرفة أم يترك فجوة الفعل والوكالة؟	بحث نظري + تجربة مصغرة

12 مشروعا قابلا للتنفيذ 1

البرنامج التنفيذي يبدأ من kernel لا من منصة ضخمة. الهدف أن تثبت ثلاث وحدات فرقا مقاسا قبل بناء نظام واسع.

كل مشروع ينتج شيئين: artifact يعمل، وnegative results. الفشل الموثق في هذا البرنامج ليس عيبا؛ إنه مادة ذاكرة وإصلاح.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع 12 مشروعا قابلا للتنفيذ 1 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

رمز	المشروع	زمن/حجم أولي
P1	Prototype MCD Router في Python مع scoring traces.	4-8 أسابيع prototype
P2	بناء Failure Atlas لخمسة benchmarks.	4-8 أسابيع prototype
P3	GraphRAG proof carrying على corpus علمي صغير.	4-8 أسابيع prototype
P4	Tool-Risk Firewall لواجهات MCP المحلية.	4-8 أسابيع prototype

12 مشروعا قابلا للتنفيذ 2

البرنامج التنفيذي يبدأ من kernel لا من منصة ضخمة. الهدف أن تثبت ثلاث وحدات فرقا مقاسا قبل بناء نظام واسع.

كل مشروع ينتج شيئين: artifact يعمل، وnegative results. الفشل الموثق في هذا البرنامج ليس عيبا؛ إنه مادة ذاكرة وإصلاح.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع 12 مشروعا قابلا للتنفيذ 2 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

رمز	المشروع	زمن/حجم أولي
P5	Benchmark Mutation Foundry لمهام GAIA-like.	4-8 أسابيع prototype
P6	Self-Reconstruction Memory Log مع expiry policies.	4-8 أسابيع prototype
P7	Skill Graph Compiler من traces ناجحة.	4-8 أسابيع prototype
P8	Perturbation Court متعدد الوكلاء.	4-8 أسابيع prototype

12 مشروعا قابلا للتنفيذ 3

البرنامج التنفيذي يبدأ من kernel لا من منصة ضخمة. الهدف أن تثبت ثلاث وحدات فرقا مقاسا قبل بناء نظام واسع.

كل مشروع ينتج شيئين: artifact يعمل، وnegative results. الفشل الموثق في هذا البرنامج ليس عيبا؛ إنه مادة ذاكرة وإصلاح.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب محكمة الهدم والبرنامج البحثي لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع 12 مشروعا قابلا للتنفيذ 3 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

رمز	المشروع	زمن/حجم أولي
P9	World-Effect Loop للتطبيق البرمجي.	4-8 أسابيع prototype
P10	Causal Map Compiler عربي/إنجليزي.	4-8 أسابيع prototype
P11	Novelty Survivability Scorer بخبرة بشرية.	4-8 أسابيع prototype
P12	MCAI Kernel dashboard للقياس والتتبع.	4-8 أسابيع prototype

قاموس مصطلحات تقني 2

القاموس هنا ليس شرح كلمات؛ إنه ضبط حدود. المصطلح يبقى فقط إذا ساعد على تحديد state أو action أو verifier أو risk أو research أو test.

أي مصطلح لا ينتج قرارا أو قياسا أو حدا يجب أن يحذف من هندسة AGI. اللسان العلمي القوي لا يكثر الأسماء؛ يكثر الفروق المنتجة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ملاحق لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع قاموس مصطلحات تقني 2 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الآلية، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

مصطلح	حد تشغيلي
Self-Reconstruction Memory	ذاكرة لا تخزن جوابا فقط بل سبب نجاحه وفشله وحدوده.
World-Effect Loop	ربط التفكير بتأثير قابل للرصد: كود يعمل، تجربة، قرار، قياس.
Benchmark Mutation	تحويل الاختبار إلى عائلة مقاومة للحفظ والتلوث والتكيّف السطحي.
Δ-Extractor	يستخرج الفروق النشطة: التناقض، النقص، التحول، اللايقين، المخاطر.
MCD-Router	يرتب المسارات بحسب فرق التعقيد: دليل، تفاعل، عمق زمني، وكالة، ضغط برهاني.

قاموس مصطلحات تقني 3

القاموس هنا ليس شرح كلمات؛ إنه ضبط حدود. المصطلح يبقى فقط إذا ساعد على تحديد state أو action أو verifier أو risk أو research أو test.

أي مصطلح لا ينتج قرارا أو قياسا أو حدا يجب أن يحذف من هندسة AGI. اللسان العلمي القوي لا يكثر الأسماء؛ يكثر الفروق المنتجة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ملاحق لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع قاموس مصطلحات تقني 3 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين القياس، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

مصطلح	حد تشغيلي
Vascular Compute Allocator	يوزع الميزانية بين بحث، تفكير، أدوات، محاكاة، محكمة نقد.
Evidence Rank Matrix	يفصل الثابت/الراجح/الخلافي/الفرضية/الاختراع/التأمل.
Causal Map Compiler	يحول النص إلى causal graph: عقد، سهام، قيود، استثناءات.
Hypothesis Reactor	يولد فرضيات من negative space وinversion وcross-domain fusion.
Perturbation Court	يعرض المخرجات لهجمات: مصدرية، سببية، هندسية، أمنية، قياسية.

قاموس مصطلحات تقني 4

القاموس هنا ليس شرح كلمات؛ إنه ضبط حدود. المصطلح يبقى فقط إذا ساعد على تحديد state أو action أو verifier أو risk أو research أو test.

أي مصطلح لا ينتج قرارا أو قياسا أو حدا يجب أن يحذف من هندسة AGI. اللسان العلمي القوي لا يكثر الأسماء؛ يكثر الفروق المنتجة.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ملاحق لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع قاموس مصطلحات تقني 4 نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين المحكمة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

مصطلح	حد تشغيلي
Self-Reconstruction Memory	يسجل أسباب النجاح والفشل لا النتائج فقط.
Moussaif Selective Assistance	منهج انتقاء مسارات مساعدة لا تعطي كل شيء بل ما يرفع القدرة الذاتية للنظام أو القارئ.
HAKIM-MCD	معمار يضع الحاكمية ومقياس فرق التعقيد في قلب الوكالة الذكية.

النموذج الجديد في صفحة واحدة: MCAI-v2 neoAGI Kernel

تعريف: MCAI-v2 neoAGI Kernel هو scaffold حاكمي فوق LLMs يجعل الذكاء مسارا لا جوابا: فرق Δ، قياس MCD، توجيه وعائي، أدوات محكومة، محكمة هدم، أثر عالمي، وذاكرة إعادة بناء.

لماذا نحتاجه؟ لأن frontier LLMs تحسن الجواب لكنها لا تضمن الحاكمية: قد تعرف كثيرا وتتصرف خطأ، أو تستخدم الأداة بلا صلاحية، أو تحفظ فشلا في الذاكرة. AGI يتطلب constitutional agency لا fluency فقط.

طبقاته: Source Ledger, Evidence Ranks, Difference Engine, MCD Router, Tool-Risk Firewall, GraphRAG, Reasoning Search, Perturbation Court, Agency Selector, World-Effect Loop, Memory Reconstruction, Benchmark Mutation.

ما يفسره أفضل: لماذا تنجح تقنيات منفصلة في benchmarks محددة وتفشل في الأفق الطويل. ما لا يفسره: كيف تتولد الفهم الذاتي أو الوعي الحقيقي؛ هذه تبقى خارج الادعاء التجريبي للكتاب.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب النموذج الجديد لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع النموذج الجديد في صفحة واحدة: MCAI-v2 neoAGI Kernel نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأثر، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **صيغة الاختبار:** إذا لم يزد kernel الاستقرار عبر 13 benchmark mutated، فهو لا يستحق اسم neoAGI.

جدول الثابت/الراجح/الخلافي/الفرضيات

هذا الجدول هو صمام الأمان النهائي. لا يسمح للكتاب أن يبيع فرضية كحقيقة أو أن يخنق الابتكار بحجة عدم وجود إجماع.

يجب تحديث هذا الجدول بعد كل دورة benchmark حقيقية. المعرفة الحية لا تخاف من تغيير رتبة أفكارها.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ملاحق لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع جدول الثابت/الراجح/الخلافي/الفرضيات نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الذاكرة، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

الطبقة	أمثلة داخل الكتاب	وضعها
ثابت/قوي	benchmarks، reasoning-action، tool-use، RAG، واقعية	مستند إلى مصادر منشورة
راجح	التحقق والذاكرة المنظمة يحسنان مهاما معينة	مشروط بالنظام والبيئة
خلافي	هل orchestration يكفي للتقدم نحو AGI؟	نزاع مفتوح
تركيب	ربط hyperskills بـbenchmark mutation	استنتاج من فجوات متعددة
فرضية/اختراع	MCD Router, Vascular Cortex OS, Inventive Entropy Governance	يحتاج prototypes
تأمل	الوعي الوعائي كعدسة للذكاء الاصطناعي	ليس ادعاء علميا مثبتا

سجل البناء المعرفي

المصادر التي غيرت اتجاه الكتاب: GAIA وWebArena وSWE-bench نقلت التركيز من المعرفة المغلقة إلى الفعل طويل الأفق؛ Reflexion وVoyager نقلا التركيز من الجواب إلى ذاكرة الفشل ومكتبة المهارات؛ MCP security نقل التركيز إلى صلاحيات الأدوات.

الفرضيات المرفوضة: أن benchmark score واحد يكفي؛ أن RAG يحل المعرفة؛ أن prompt طويل يحل الحاكمية؛ أن novelty وحدها قيمة؛ وأن AGI يمكن أن يعلن من غير trace تشغيلية.

النموذج الأولي تحسن من «نموذج مفكر» إلى «نظام حاكمي»: لم يعد LLM هو البطل الوحيد، بل صار عضوا داخل vascular governance.

أقوى نقطة قابلة للنقد: MCD يحتاج تشغيل كمي دقيق وأوزانا قابلة للتعلم. أقوى إضافة: تحويل الإبداع إلى أدوات محكومة ومقاييس نجاة لا إلى أسماء.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ملاحق لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع سجل البناء المعرفي نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الأمن، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **منطقة جهل:** لا نعرف بعد هل هذا scaffold سيصمد عبر نماذج مختلفة وتوزيعات واقعية؛ هذا هو مشروع الاختبار التالي.

اختبار الجودة النهائي

التقييم الداخلي من 100 ليس تزكية نهائية بل فحص كتابي قبل التسليم. الدرجة تقيس البنية والكثافة والصدق الرتبي لا صحة AGI.

نقاط القوة: دمج المصادر الحديثة بالمنهج المساعفي، فصل المحاكاة عن النتائج الرسمية، بناء أدوات قابلة للبروتوتايب، ومحكمة هدم صريحة. نقاط الضعف: بعض الأدوات تحتاج formal specs أكثر، وبعض الأرقام محاكاة لا تنفيذ.

الحكم: الكتاب جاهز كنسخة بحثية هندسية v3، لكنه ليس ورقة تجريبية. المرحلة التالية: تنفيذ ثلاثة prototypes وقياسها على subset من GAIA/SWE-bench/WebArena/ARC-like tasks.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب ملاحق لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع اختبار الجودة النهائي نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين النقل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

المعيار	النقطة
عمق التأصيل	14/15
جودة المصادر	14/15
كثافة الصفحات	9/10
وضوح البنية	9/10
قوة النقد	9/10
الأصالة والتركيب	14/15
قابلية الاختبار	9/10
جمال PDF	4/5
عدم الحشو	4/5
فائدة القارئ المتقدم	5/5
المجموع	91/100

خاتمة كثيفة: ما الذي أضفناه فعلا؟

أضفنا تحولا: من سؤال «أي نموذج أقوى؟» إلى سؤال «أي نظام حاكمي يجعل القوة قابلة للفحص والإصلاح؟». هذا التحول هو لب MCAI-v2/ neoAGI-v3.

أضفنا ثلاث نظريات لا كيقين بل كمحركات بحث: الحاكمية الوعائية السببية، نواة الهايبرسكل، وحاكمية الإنتروبيا الاختراعية. وأضفنا ثلاث تقنيات: Proof-Carrying HyperGraphRAG، Vascular Cortex OS، وBenchmark Mutation Foundry.

أضفنا 15 أداة تقنية قابلة للتحويل إلى prototypes، وخمس فلسفات تدريب، وخمس أدوات تدريب، ومحاكاة 13 benchmark مع فصل صريح بين الهدف النظري والنتيجة الرسمية.

الكتاب لا يقول إن AGI تحقق. يقول إن الطريق الأكثر جدية ليس تضخيم الإجابات بل بناء حاكمية للفروق والأدلة والأدوات والذاكرة والفشل. الذكاء العام إن اقترب، فسيقترب كنظام يتعلم كيف يحاكم نفسه لا كنص يمدح نفسه.

تشغيل الصفحة: تُقرأ هذه الصفحة كعقدة داخل الباب خاتمة لا كمعلومة منفردة. السؤال العملي هو: ما المدخل الذي تستقبله، ما القرار الذي تغيره، وما الأثر الذي تتركه في مسار MCAI-v2؟ عند تحويلها إلى نظام، لا يسمح لها بأن تبقى وصفا؛ يجب أن تنتج حقل JSON أو verifier أو قاعدة route أو سجل فشل.

تمييز معرفي: داخل موضوع خاتمة كثيفة: ما الذي أضفناه فعلا؟ نفصل بين ما يمكن نسبته إلى الأدبيات وما هو تركيب مساعفي. الأدبيات تعطينا شواهد جزئية، أما التركيب فيبني علاقة بين الدليل، الفعل، والذاكرة. هذا الفصل يمنع الانزلاق من «مصدر قال» إلى «نظام يضمن»؛ الضمان لا يأتي إلا بعد benchmark متحور وأبلشن.

تنزيل تقني: أبسط صيغة تنفيذية هي أن نجعل كل مفهوم بطاقة تشغيل: definition, assumptions, inputs, outputs, failure modes, metrics, attacks, repair rule. بهذه الصيغة تتحول الفصول إلى مكتبة هندسة لا إلى نص طويل. وإذا عجز المفهوم عن ملء هذه البطاقة فهو إما ناقص أو زخرفي.

محكمة مصغرة: اسأل دائما: ما الفرضية الخفية هنا؟ ما الحالة القصوى التي تكسرها؟ أي مدرسة منافسة ستقول إن هذا مجرد إعادة تسمية؟ ما التجربة التي تفرق بينه وبين baseline؟ الجواب المقبول لا يكتفي بالدفاع؛ يعيد تصميم المفهوم حتى يصير أضيق وأقوى.

مؤشر قياس: الصفحة الناجحة ترفع واحدا من خمسة: دقة الحكم، استقرار المسار، قابلية الاسترجاع، أمان الأداة، أو جودة novelty. ما لا يرفع شيئا منها يظل مفيدا أدبيا لكنه لا يدخل kernel. هكذا تتحول الكثافة إلى اقتصاد معرفي لا إلى تراكم كلمات.

ربط بالنظام العام: المسار الكامل يبقى ثابتا: فرق نشط Δ، تقدير MCD، توجيه وعائي، استدعاء مصدر أو أداة، محكمة اضطراب، اختيار وكالة، أثر عالمي، وذاكرة إعادة بناء. كل صفحة في الكتاب يجب أن تحدد موضعها في هذا المسار وإلا فقدت وظيفة AGI engineering.

مصفوفة تحويلية: حوّل الفكرة إلى أربعة أصول: rule مختصرة، protocol قابل للتنفيذ، benchmark fragment يقيسها، وfailure note يصف كيف تخدعنا. عند وجود هذه الأصول الأربعة تصبح الصفحة مادة تدريب وقياس، لا مادة قراءة فقط.

حالة قصوى: افترض أن المصدر ناقص، الأداة مضللة، السياق طويل، والنتيجة مرغوبة عاطفيا. هل يبقى الحكم صحيحا؟ إذا انهار، فالمشكلة ليست في النموذج وحده بل في غياب حاكمية تمنع المسار السهل من انتصار كاذب.

خطة تطوير: النسخة الأولى تصف المفهوم؛ الثانية تضيف schema؛ الثالثة تربطه بtrace؛ الرابعة تقيسه بأبلشن؛ الخامسة تضيف mutation. لا تنتقل إلى كلمة AGI قبل المرور بهذه السلسلة؛ وإلا أصبح المشروع تسمية مستقبلية بلا نظام اختبار.

تمييز ضد الحشو: التكرار المقبول هو تكرار بروتوكولي يثبت معيارا في سياقات مختلفة. التكرار المرفوض هو إعادة العبارة بلا اختلاف في آلية أو خطر أو تطبيق. لذلك تعاد القاعدة هنا بصيغة تشغيلية لا إنشائية.

إغلاق الصفحة: لا تخرج الفكرة من هذه الصفحة إلا ومعها سؤال بحثي واحد، وخطر واحد، ومؤشر واحد، وأقرب prototype. هذه الأربعة هي ختم MCAI: معرفة تتحول إلى قدرة، وقدرة تقبل الهدم، وهدم يولد تحسينات لا إحباطا.

> **الصفحة الأخيرة:** كل تقنية لا تنجو من محكمة الهدم تُعاد إلى المختبر، وكل فكرة تنجو بلا تنفيذ تبقى ناقصة.