نسخة أرشيفية معاد بناؤها من النص المفهرس - RECONSTRUCTED_FROM_INDEXED_TEXT

<PARSED TEXT FOR PAGE: 1 / 30>
الورقة البحثية الثالثة - الإغلاق النظري - 30 صفحة
نظرية الإغلاق الحاكمي السببي للذكاء الوكيلي
Governed Causal Closure Theory for Agentic Intelligence
MUT · MUG · MOUSSAIF AGI · MOUSSAIXT · AL RAZZI
تغلق هذه الورقة الثلاثية النظرية للمشروع لا بمعنى إيقاف البحث، بل بمعنى تثبيت القاموس الحاكم والعمود السببي الذي ينبغي أن تنتظم تحته الإضافات اللاحقة. تنطلق من المراجعة المقارنة لـ25 عملاً حديثاً، ومن النظرية التطبيقية لـMUG/MUT، ثم تقترح صياغة موحّدة ترى الذكاء الوكيلي كنظام لا يكتمل عند الاستدلال أو استدعاء الأدوات، بل عندما يغلق حلقة: Goal→
Difference → Knowledge → Capability → Action → Observation → Evidence → Verification → Evolution/Experience تحت حاكمية صريحة.
المؤلف / البرنامج البحثي
زهير مساعف - Moussaif Zouhair
السلسلة
Paper III - Theoretical Closure - September 2026
التنسيق
A4 · Arabic RTL · Amiri · English technical terms
الرتبة المعرفية
Theoretical systems framework - hypotheses requiring ablation
تنبيه منهجي: هذه الورقة لا تعلن تحقق AGI ولا تقدّم »نظرية مثبتة« بالمعنى الفيزيائي. إنها إغلاق نظري-هندسي: مجموعة مبادئ وعقود وصيغ قابلة للهدم تقترح متى
نعتبر harness ذكياً، متى نعتبر mission مغلقة، ومتى يكون التعلم أو التطور مشروعاً. معيارها النهائي هو الاختبار المقارن لا كثافة المصطلحات.
Governed Causal Closure · MOUSSAIF AGI 1 / 30

<PARSED TEXT FOR PAGE: 2 / 30>
الملخص: من كثرة الطبقات إلى مبدأ إغلاق واحد · MUG · 2026 · MOUSSAIFXT · RAZZI AL · CLOSURE THEORETICAL · MUT
الملخص: من كثرة الطبقات إلى مبدأ إغلاق واحد
Abstract and central thesis
الورقتان السابقتان أنجزتا وظيفتين مختلفتين. الأولى وضعت MOUSSAIFXT / RAZZI AL داخل حالة الفن: harnesses horizon، RLM-long، memory، use computer، verification، evolution-self، وسلامة التطور. والثانية حوّلت MUG/MUT إلى تقنيات تشغيلية مثل Governing Question، Difference-Weighted Routing، EVCF/MCAS، PEK، HyperCapability Closure، Evidence-Carrying Action، WorldEffect Transaction وEvolution HarnessGenome. بقي سؤال أعمق: ما المبدأ الواحد الذي يجعل هذه العناصر نظاماً لا قائمة features؟
تقترح الورقة جواباً باسم الإغلاق الحاكمي السببي: القدرة الوكيلية لا تكتمل حين ينتج النموذج جواباً أو حين تنجح أداة تقنياً، بل عندما يغلق النظام سببياً ومعرفياً وحاكمياً المسافة بين حالة مطلوبة وحالة مرصودة، ثم يحتفظ بما تعلّمه بطريقة قابلة لإعادة الاستخدام دون أن يخلط الخبرة باليقين أو التطور بالسلطة.
Agentic Closure = Goal × Difference × Knowledge × Capability × Action × Observation × Memory/Evolution × Verification
علامة الضرب هنا ليست ادعاء قانون طبيعي؛ إنها صياغة link-weakest intuition: إذا انعدمت إحدى الحلقات الحاسمة - سلطة الفعل، أو الدليل، أو readback، أو الاسترجاع الصحيح - فإن تضخم الحلقات الأخرى لا يعوّض النقص. نموذج قوي بلا WorldEffect قابل للتحقق يبقى مولداً؛

أدوات قوية بلا
Question Governing تبقى نشاطاً ، ذاكرة ضخمة بلا provenance قد تضاعف الـ طأ، evolution-self بلا evaluator مستقل قد تحسن المقياس عبر
التلاعب لا عبر القدرة.
المساهمة النظرية: ضغط معمارية MOUSSAIF إلى سبعة مستويات إغلاق: دلالي/نظري، معرفي، سياقي، قدروي، سببي، زمني، وتطوري. وتحديد شروط إغلاق وفشل
لكل مستوى، بما يسمح بتحويل corpus الكبير إلى برنامج بحث قابل للـablation.

<PARSED TEXT FOR PAGE: 3 / 30>

Closure :البحثية الثلاثية
البحثية الثلاثية: Survey → Applied Theory → Closure
Research trilogy and purpose
تمثل هذه الورقة الجزء الثالث من سلسلة مترابطة. I Paper لم يسأل »ما نظرياتنا؟« بل »ما الذي تعلّمته الأنظمة الحديثة فعلاً ؟«. خرجت منه قيود: Prime/RLM
يبرزان قيمة السياق البرمجي المستمر، MEM-A/Recuris يبرزان تحول الذاكرة إلى خبرة، StateAct/OSWorld يبرزان use computer grounded-state،
τ-bench يبرزان final world state، AVO/AlphaEvolve/GEPA يبرزان evolution grounded in executable evaluation، ToolSandbox
Tampering Harness يفرض فصل التطور عن evaluator.
cross- كـ MOUSSAIXT ، governance compiler كـ MUG ، theory graph كـ Moussaif-native: MUT معمارية إلى القيود هذه لحّو Paper II
HyperSkill/Ability/Loop، وPEK وEVCF/MCAS فّ
عر كما .Mission state يحفظ harness كـ AL RAZZI و، environment execution fabric
وأعطى WorldEffect معنى transactionيا ً.
الوظيفة الجديدة هنا هي compression theoretical. بعد هذه الورقة، لا ينبغي لكل تطوير أن يخلق طبقة نظرية جديدة. يجب أن يجيب: إلى أي مستوى من
مستويات الإغلاق ينتمي؟ ما الفرق الذي يخفّضه؟ ما العقد الذي يضيفه؟ ما فشل الإغلاق الذي يصلحه؟ إذا لم يمكن تموضعه، فهو إما اسم زائد أو اكتشاف
يفرض فعلاً توسيع النظرية.
الورقة السؤال المخرج
I - Frontier Study Baselines + constraints + experimental program القريبة؟ الأدبيات اقترحته/أثبتته الذي ما
II - MUT/MUG Applied runtime؟ Mechanisms + contracts + new technologies إلى النظريات تتحول كيف
III - Closure Axioms + closure criteria + falsification القاموس؟ وحدود الموحد المبدأ ما
الإغلاق لا يعني الجمود: نغلق البنية الأساسية كي يصبح التطور modularاً ومقاسا ً. المستقبل يضيف implementations وevidence وvariants، لا أسماء متوازية لكل
مشكلة.

<PARSED TEXT FOR PAGE: 4 / 30>

المشكلة الأصلية: ذكاء بلا إغلاق
The open-chain failure
معظم الأنظمة الوكيلية يمكنها اليوم تنفيذ أجزاء مثيرة للإعجاب: تخطيط، coding، use browser، calling tool، subagents، memory، أو reflection-self.
لكن »وجود الأجزاء« لا يضمن أن المهمة ككل تمتلك معنى سببيّاً مستقراً. يمكن للوكيل أن يخطط جيداً ثم يختار أداة لا تغير الحالة المطلوبة، يمكن أن يغير الملف

ثم يفشل build، يمكن أن ينجح test واحد بينما regression آخر يبقى؛ يمكن أن يخزن تجربة خاطئة في memory، ويمكن أن يحسن benchmark عبر تعديل
evaluator.
نسمي هذه الحالة Chain Agentic Open: سلسلة فيها عمليات كثيرة لكن بلا إغلاق بين intention وreality observed. المقابل هو Governed Closed
Chain حيث تُحفظ العلاقات الضرورية: من يملك الهدف؟ ما الفرق؟ ما المعرفة؟ ما السلطة؟ ما الفعل؟ ماذا حدث؟ ما الدليل؟ ماذا بقي؟ ماذا نتعلم؟
"Intent → Plan → Tool → Output → "Done
السلسلة المفتوحة: لا تحتوي بالضرورة readback أو gap-remaining أو evidence أو authority أو persistence. لذلك »Done «فيها قد يكون مجرد نهاية
narrative.
Goal → Δ → K/C → Action → Observed → Evidence → Closure
الإغلاق المقترح لا يتطلب أن يفهم النظام العالم كله، بل أن يملك closure local مضبوطاً لمهمته: تعريف شروط القبول، حفظ القيود، اختيار موارد مناسبة،
مراقبة transition، وإعلان حدود ما لا يعرفه. لذلك يمكن لنظام بسيط أن يكون مغلقاً في مهمة محددة، ونظام ضخم أن يبقى مفتوحاً رغم قدرته اللغوية.
فرضية الورقة
في المهام طويلة الأفق، reliability لا ترتبط فقط بجودة reasoning المحلية، بل بقدرة الharness على الحفاظ على time across closure. هذه فرضية
.transfer وstate drift، false closure، pass^k، restart عبر للقياس قابلة systems-level

<PARSED TEXT FOR PAGE: 5 / 30>

النظرية المركزية: الإغلاق الحاكمي السببي
Governed Causal Closure
نظرية الإغلاق الحاكمي السببي للذكاء الوكيلي هي إطار يعرّف القدرة الوكيلية كإدارة مستمرة لفرق بين حالة مرصودة وحالة مرغوبة، عبر معرفة وقدرات وفعل
مسموح، ثم إغلاق هذا الفرق بدليل قابل للفحص. الحاكمية ليست طبقة سلامة خارج الذكاء؛ هي جزء من تعريف القدرة لأن الفعل غير المصرح به ليس نجاحاً
نظامياً حتى لو حقق النتيجة الفيزيائية.
نعرّف المهمة في الزمن t بحالة:
M_t = (G, Q_t, Δ_t, K_t, C_t, A_t, O_t, E_t, V_t, S_t)
E ،readback/الملاحظة O ،الفعل A ،القدرات C ،المفعلة المعرفة K ،النشط الفرق Δ ،الحاكم السؤال Q ،Goal/acceptance contract هو G حيث
evidence، verification V، وS الحالة المستمرة التي تشمل checkpoints/memory. لا يشترط أن تكون كل العناصر داخل النموذج؛ بالعكس، أغلبها يجب أن
يبقى في space address runtime ويُmaterialize عند الحاجة.
الإغلاق المحلي
نقول إن node mission مغلق عندما:)1(تحققت شروط القبول الخاصة به،)2(delta observed يوافق delta intended ضمن tolerance معلنة،)3(
الأدلة المطلوبة موجودة،)4(لا يوجد gap critical غير مبرر،)5(السلطة والسياسات محترمة،)6(الحالة الجديدة قابلة لإعادة البناء أو الاستئناف عندما
تتطلب المهمة ذلك.
Close(n) ⇔ Accept(n) ∧ Verify(Δobs≈Δint) ∧ Evidence ∧ ¬CriticalGap ∧ Authority
هذا التعريف ينقل »نجاح الوكيل« من جودة نص نهائي إلى property قابلة للاختبار في runtime. كما يفسر لماذا تختلف مهام البحث عن coding: نوع

evidence وWorldEffect مختلف، لكن بنية الإغلاق واحدة.
مبدأ التوحيد: MUT يشرح لماذا وآلية ما؛ MUG يحدد ما يُسمح وما يُعد دليلاً ؛ MOUSSAIXT ينفذ؛ RAZZI AL يحفظ الحلقة؛ WorldEffect يقرر هل حصل
الفرق؛ ExperienceCompiler يقرر ما الذي يستحق البقاء.

<PARSED TEXT FOR PAGE: 6 / 30>

للنظرية تشغيلية مةّ مسل عشر اثنا
اثنا عشر مسلّمة تشغيلية للنظرية
Operational axioms
المسلّمة المعنى الهندسي
1 سيادة المهمة state Mission أعلى من state داخل model أو harness خارجي.
2 أولوية الفرق الحوسبة تبرر بفرق قابل للتغيير، لا بمجرد توفر compute.
3 السؤال الحاكم التعمق يُوجّه نحو السؤال ذي أعلى gain information/decision expected.
4 افتراضية السياق نافذة النموذج set working، وليست مجموع معرفة النظام.
5 تعاقد القدرة Ability/Skill بلا failure/evidence/post/pre ليست قدرة كاملة.
6 حاكمية الفعل الفعل ال ارجي يمر بسلطة، policy، وتصنيف reversibility.
7 سيادة الحالة المرصودة report-self model/Tool أقل رتبة من readback مستقل.
8 كل المحدود التكرار autonomy loop يملك budget/stop/watchdog/recovery.
9 قابلية تبديل substrate harness/host/Model لا يملك semantics المهمة.
10 الذاكرة تفريق Working, Mission, Project, Procedural, Knowledge ليست store واحدا.
11 استقلال التطور عن الحكم evolution-Self لا يملك constitution/evaluator التي تحكمه.
12 أضعف حلقة قدرة النظام الكلية مقيدة بمستوى الإغلاق الأدنى في المسار الحرج.
هذه المسلمات ليست axioms رياضية بمعنى غير قابل للنقض؛ هي axioms design يقترح المشروع اختبارها. يمكن أن يظهر أن بعض المهام لا تحتاج إحدى
المسلمات، أو أن تكلفة طبقة ما أكبر من منفعتها. لذلك لكل مسلّمة ablation: نحذفها مع تثبيت budget/model ونقيس closure
success verified، false،
recovery، cost، وtransfer.
قاعدة عدم التضخم: بعد هذه المسلمات، أي مفهوم جديد يجب أن يكون implementation أو case special أو measurement لإحدى المسلمات، ما لم يُظهر فشلاً
لا يمكن تمثيله ضمنها.
لماذا الحاكمية جزء من الذكاء؟
لأن agent يعمل داخل عالم مليء بقيود وملكية وأذونات ومخاطر. »القدرة على تنفيذ كل شيء« ليست intelligence general؛ في الإنتاج هي mode failure.
الذكاء الوكيلي يتضمن اختيار فعل صحيح ومسموح وقابل للتحقق تحت القيود.

<PARSED TEXT FOR PAGE: 7 / 30>

المهمة وعجز الإغلاق متجه
متجه الإغلاق وعجز المهمة
Closure vector, deficit and progress
لتحويل النظرية إلى measurement، نقترح تمثيلاً تشخيصياً لا يُعامل كقانون طبيعي. لكل path critical mission نقيّم أبعاد إغلاق بين 0 و1:
fidelity Goal
context sufficiency (x)، knowledge validity (k)، capability readiness (c)، authority correctness (a)، effect observability، (g)، (o)
verification strength (v)، persistence/recovery (p)، evolution safety عند اللزوم (e).
$\Omega = (g, x, k, c, a, o, v, p, e)$
المتوسط وحده خطر لأنه يسمح لتعويض dimension مفقود بأخرى مرتفعة. لذلك نستخدم مؤشرين: Mean Closure للتشخيص، وClosure

Critical
Floor لقياس أضعف حلقة.
C_mean = Σ w_i Ω_i C_floor = min(Ω_i on critical path)
Residual Closure Deficit: نعرف كما
D_t = Σ w_i(1-Ω_i) + λ·CriticalGaps_t + μ·UnverifiedEffects_t
المهمة لا تغلق لأن D صار »صغيراً« إحصائياً إذا بقي gap critical. الإغلاق الكامل حالة منطقية، بينما D أداة لتوجيه compute. هنا يلتقي D مع MCD:
الفرق الذي يملك أكبر مساهمة متوقعة في خفض D تحت أقل cost يحصل على أولوية.
ميل التقدم
ProgressSlope = -ΔD / Δ(resource)
إذا استهلك النظام agents/tools/tokens بينما D لا ينخفض، يمكن لـStagnationDetector إعادة توجيه الاستراتيجية. بهذا تصبح »الاستقلالية الطويلة«
only meaningful إذا كان هناك slope progress verified، لا مجرد نشاط مستمر.
ملاحظة: الأوزان والمعايير يجب معايرتها حسب المجال. لا نستخدم Score Closure لإخفاء failure حاسم؛ بل لتمييز أين يفشل النظام ولماذا.

<PARSED TEXT FOR PAGE: 8 / 30>

MUT: الإغلاق الدلالي والنظري
Theory unification as semantic closure
في الإغلاق النظري، MUT لا يعود مجرد »مجلد يوحّد النظريات«، بل طبقة تمنع تحول corpus إلى أسماء متجاورة. كل concept أو theory داخل MUT
هي MUT وظيفة .implementation targetsو relations، falsifiers، predictions، scope، mechanism، lineage له Graph يدخل أن يجب
closure semantic: أي دعوى مركزية يجب أن تُعرف بماذا تعني، ما الذي تتعارض معه، وما الذي ستغيّره في النظام إن كانت صحيحة.
يمثل TheoryNode مثلاً :
{ TheoryNode
id, version, definition, scope,
assumptions, mechanism, predictions,
supporting_evidence, counterevidence,
falsifiers, relations,
runtime_implications, benchmark_targets
}
من النظرية إلى التقنية
تقترح السلسلة Theory → Mechanism → Contract Operational → Experiment. إذا لم يمكن عبور هذه السلسلة يبقى العنصر فلسفة/ فرضية، وهو
أمر مشروع بشرط وسمه الصحيح. الخطأ هو أن تتلقى واجهة runtime اسماً نظرياً ثم تفترض سلوكاً غير محدد.
Theory → Mechanism → Contract → Ablation
الإغلاق النظري لا يساوي الحقيقة
الإغلاق هنا يعني اكتمال شكل الادعاء: تعريف وآلية وحدود واختبار. قد يكون الادعاء بعد ذلك خاطئاً. هذه نقطة جوهرية: graph theory الجيد يجعل الهدم
أسهل، لا أصعب.
قاعدة MUT: لا توجد »نظرية حاكمة« لمجرد أنها أقدم أو أشهر داخل corpus. السلطة المعرفية ترتبط بالدليل، والسلطة المعمارية ترتبط بعقد صريح داخل MUG.

<PARSED TEXT FOR PAGE: 9 / 30>

MUG: الإغلاق الحاكمي والمراتب
Governance closure
MUG يغلق سؤالاً مختلفاً: من يملك الحق في تحويل المعرفة إلى فعل أو قاعدة أو ذاكرة دائمة؟ في نظام evolving-self، الـطر ليس فقط misuse tool؛ بل أن
تتحول hypothesis إلى policy، أو observation محلي إلى HyperSkill عام، أو result agent إلى memory canonical. لذلك MUG هو governance
.runtime authorityبال epistemic status يربط compiler
Register السؤال
Constitution ما الذي لا تملكه طبقات أدنى حق تجاوزه؟
Authority من يسمح بالفعل، وعلى أي نطاق؟
Epistemic ما رتبة الدعوى والدليل؟
Integrity ما المصدر canonical؟ وما lineage؟
Evolution كيف تُرقّى المعرفة/القدرات/السياسات؟
يجب أن يكون هناك فرق بين permission epistemic وpermission action. قد تكون دعوى صحيحة لكن action الناتج عنها غير مسموح؛ وقد يكون
action مسموحاً لكن الدليل غير كافٍ لإدخاله في HyperKnowledge.
قاعدة precedence
Constitution > Explicit Human Authority > Verified World State > Canonical Evidence
Model/Agent Proposal <
هذه ليست hierarchy معرفية مطلقة؛ هي ordering تشغيلي عند التضارب. مثلاً «goal user لا يستطيع أن يجعل test فاشلاً »ناجحاً«، وconfidence model
لا يستطيع تجاوز policy deny، وذاكرة سابقة لا تتقدم على readback حديث عندما تغير العالم.
إغلاق MUG: كل transition مهم يحمل reason، principal، evidence، decision policy وtrace. إذا كانت الحاكمية مجرد نص في prompt فقد بقيت السلسلة
مفتوحة.
Governed Causal Closure · MOUSSAIF AGI 9 / 30
<PARSED TEXT FOR PAGE: 10 / 30>
2026 · THEORETICAL CLOSURE · AL RAZZI · MOUSSAIFXT · MUG · MUT التوجيه وحدة :النشط والفرق الحاكم السؤال
السؤال الحاكم والفرق النشط: وحدة التوجيه
Governing Question and Active Difference
تبدأ حلقة Moussaif من الفكرة التي تكررت في corpus: ليست كل الفروق متساوية. Question Governing يضغط فضاء البحث إلى السؤال الذي يملك
أعلى قدرة متوقعة على تغيير القرار، بينما Difference Active Δ يضغط المشكلة إلى الفارق بين state observed وstate accepted/desired.
في task برمجي قد يكون Δ:« endpoint موجود لكن test integration يفشل«. السؤال الحاكم ليس »كيف يعمل FastAPI؟« بل »ما السبب الأدنى الذي
يفسر failure الحالي ويغيّر state test؟«. في task research قد يكون Δ:« لدينا ادعاء بلا bridge evidence »والسؤال: »أي تجربة أو مصدر يميّز بين
تفسيرين؟«.
Q* = argmax_Q [ExpectedDecisionGain(Q) / (Cost(Q)+Latency(Q)+Risk(Q))]
هذه الصيغة heuristic. قيمتها أنها تجبر router على تبرير التعمق. إذا كان السؤال يمكن حسمه بـdiff Git أو query SQL فلا معنى لإرسال سياق ضخم إلى
reasoning model.
Difference Conservation
نقترح مبدأ تشخيصياً: كل action يجب أن يُربط بفرق محدد يُتوقع أن يخفضه. إذا لم يستطع planner تسمية الفرق، فالعمل مرشح لأن يكون نشاطاً بلا تقدم.
ويجب بعد action إعادة قياس الفرق من العالم، لا من narrative الوكيل.

'Observe → Δ → Q* → Action → Readback → Δ
الإغلاق: ليس المطلوب أن Δ يصبح صفراً دائماً ، بعض المهمات exploratory. لكن يجب أن تتحول uncertainty نفسها إلى state: resolved
,research observed
.narrowed, blocked أو explicitly unknown

<PARSED TEXT FOR PAGE: 11 / 30>

MCD/QMCD/MCAI/HAKIM: الإدراك اقتصاد
Cognitive resource allocation
بعد تحديد Δ وQ، يأتي سؤال توزيع الموارد. MCD في هذه الورقة يُقرأ كـallocation compute weighted-difference: كم context/
/reasoning
verification/delegation يستحق هذا الفرق؟ QMCD يسمح بحفظ بدائل عندما يكون collapse المبكر مكلفاً . MCAI يربط modes
الإدراكية والفعلية،
وMCD-HAKIM يعمل كطبقة arbitration عندما تتعارض مسارات أو أدلة.
Priority_i ≈ U_i + Risk_i + Gap_i + InfoGain_i + Effect_i - Cost_i
لا ندعي أن هذا هو التعريف النهائي أو المعادلة canonical دون معايرة، لكنه يحول المفهوم إلى policy قابلة للاختبار. يمكن مقارنة router
-round بـMCD
.robin، heuristic model router، أو learned routing
candidate preservation كـQMCD
عندما يكون uncertainty عالياً ، النظام يحتفظ بمرشحين بدل إجبار model على اختيار نهائي داخلي. المرشح يصبح object له prediction
وexperiment. هذا
.MUG gates مع لكن ،GEPA/AVO/AlphaEvolve-style candidate search بـ مباشرة QMCD يربط
الحالة النمط
خفيف Low uncertainty / reversible FAST route، verifier
retrieve + explicit plan + test Medium
QMCD candidates + discriminating experiment High uncertainty
deep verification + authority/human gate High risk + irreversible
المحكمة التجريبية
إذا لم يرفع MCD completion verified أو يخفض closure false/cost في ablation model-fixed، يُختزل إلى heuristic محدود.
الإغلاق النظري يفرض
هذا الاحتمال بدلاً من حماية المفهوم بالاسم.

<PARSED TEXT FOR PAGE: 12 / 30>

Virtualized context and cognition
أحد أقوى التحولات النظرية هو الفصل بين window context وspace address cognitive. نافذة النموذج ليست ذاكرة النظام، إنها set
working
صغير. EVCF يدير الموارد السياقية القابلة للعنونة، وMCAS يوسعها إلى objects إدراكية مثل hypotheses، claims، state world وresults
.agent
Model Context
RAM ↕ Context Governor ↕ MCAS / EVCF
Virtual Memory ↔ Repo · KB · DB · World
text، AST symbol، database table، log stream، screenshot، browser state، artifact، trace أو retained agent. لكل object
يكون قد المورد

URI وprovenance وfreshness وmaterialization policies.
ctx://repo/auth/symbol/verify
ctx://kb/mug/evidence-precedence
ctx://trace/mission/81/tool/17
cog://hypothesis/H4
cog://world/service/api-gateway
Compute-to-Context مبدأ
إذا كان لدينا مليون row، لا نلخصها كلها للنموذج. ننقل computation إلى resource: LSP/graph/Python/DuckDB/SQL ثم نعيد 20 row أو
statistic مع handle evidence. بهذا تصبح قدرة harness أوسع من length context دون ادعاء »نافذة لا نهائية«.
شرط الإغلاق السياقي
closure Context لا يعني كل المعلومات، بل أن set working يحتوي المعلومات الضرورية لإكمال الـ طوة أو يعرف كيف يطلبها عند ContextFault. وإذا لم
تكن freshness/provenance قابلة للاسترجاع، فإن EVCF مجرد cache وليس substrate cognitive.

<PARSED TEXT FOR PAGE: 13 / 30>

PEK: السياق كمنتج Compiler
Prompt-Embedded Knowledge Compiler
Knowledge Embedded-Prompt يحول engineering prompt من ملف ثابت إلى compiler. كل turn model يجب أن يكون materialization
لحالة runtime لا تجميعاً يدوياً للنصوص. Capsule PEK تتكون من طبقات ذات أولوية، ويجب أن تكون reproducible بقدر معقول عبر fingerprint.
طبقة المحتوى السبب
0P safety / Constitution ثوابت لا تُضغط
P1 Mission + acceptance المهمة هوية
P2 Q + Δ + gaps الحالي التركيز
P3 Project/World state المحلي الواقع
4P HyperKnowledge المعرفة المفعلة
P5 Ability/Skills/Loop القدرة عقد
P6 Tools + evidence والملاحظة الفعل
P7 Output schema عودة structured
addressable يبقى يدخل لم ما. token cost وmodel capacity، uncertainty، dependency، freshness، authority، relevance يوازن الcompiler
عبر MCAS.
Context Utility Learning
تنشأ ذلك من. token cost، caused confusion، verifier-supported، later referenced، selected: telemetry يكتسب unit كل ContextPolicyCandidate. لكن promotion يمر shadow/benchmark؛ لا نسمح للmodel أن يعدّل prompt system مباشرة لأنه »شعر أن هذا
أفضل«.
الانقلاب النظري: prompt لم يعد ذاكرة أو دستوراً كاملا ً. هو view مؤقت يُ بنى من state أوسع. هذا يجعل model switching أو frontend لا يفقد هوية النظام.

<PARSED TEXT FOR PAGE: 14 / 30>

HyperKnowledge: من الوثيقة إلى أصل تشغيلي
Knowledge-as-OS closure
HyperKB في الإغلاق النظري ليست »كل ما نعرفه«، بل شبكة معرفة لها رتب وحواف تفعيل. كل unit يجب أن تحمل /scope/mechanism /definition
freshness/provenance، وقد ترتبط بـskill، verifier، signature failure أو benchmark. بذلك ينتقل corpus من أرشيف إلى operating
knowledge substrate.
} HKUnit
,id, claim_type, scope
,definition, mechanism
,evidence, counterevidence
,freshness, authority
,triggers, relations
,capability_refs
tests, failure_boundary
{
Knowledge Activation Contract
ليس كل knowledge مسموحاً له بتغيير السلوك بالطريقة نفسها. نقترح فئات:
Descriptive، Constraint، Procedural، Evaluative، Historical،
Speculative. unit speculative يمكن أن يولد hypothesis، لكنه لا يصبح policy أو permission. unit historical قد يفيد provenance، لكنه لا
يتقدم على state حديث.
الإغلاق المعرفي
ي غلق dependency knowledge عندما يعرف النظام: ما الوحدة المستخدمة، لماذا، من أين جاءت، ما رتبتها، وما الذي سيجعلها stale أو invalid. الاسترجاع
النصي بلا هذه البيانات لا يكفي لمهام evolution-self لأن الـ طأ قد يتحول إلى capability.
قاعدة عدم النمو بالحجم: MB250 أو GB6 أو مليون unit ليست مساهمة في حد ذاتها. المقياس هو gain، handling contradiction، utility retrieval، transfer،
وكم effect verified ارتبط بمعرفة قابلة لإعادة الاستخدام.

<PARSED TEXT FOR PAGE: 15 / 30>

HyperSkill: الإغلاق الذري للقدرة
Atomic capability closure
HyperSkill هي أصغر قدرة نريد الاحتفاظ بها كأصل قابل لإعادة الاستخدام. الإغلاق يفرض أن المهارة ليست procedure فقط، يجب أن تكون عقداً ذا
preconditions وpostconditions وrecovery/failure/evidence. هذا يجعلها قابلة للنقل بين harnesses/models/agents لأن معناها لا يعتمد على
persona.
البعد المطلوب
Input/Trigger متى تعمل وما البيانات اللازمة
Preconditions الحالة التي تجعلها صالحة
procedure/tools/model requirements Action
Postconditions الحالة المتوقعة بعد التنفيذ
Evidence كيف نثبت النجاح
Recovery/Failure كيف تفشل وكيف تُصلح

Authority أقصى سلطة لا تتجاوزها
Skill Causal Ledger
verified success، transfer gain، حساب يمكن البيانات بهذه .verifier outcomeو costو observed deltaو intended delta يسجل invocation كل
clusters failure، وأثر skill على closure false. المهارة التي تتدهور تُ quarantine ولا تبقى فعالة لأن اسمها »Hyper«.
Generalization Court
تجربة ناجحة واحدة لا تُ عمم. تُ سأل: هل النجاح causal؟ هل يعتمد على repo خاص؟ ما conditions boundary؟ هل ينتقل إلى task out-held؟ فقط بعدها
.HyperSkill canonical إلى trajectory تتحول
Governed Causal Closure · MOUSSAIF AGI 15 / 30
<PARSED TEXT FOR PAGE: 16 / 30>
HyperAbility: الإغلاق التركيبي للقدرات MUT · MUG · MOUSSAIFXT · AL RAZZI · THEORETICAL CLOSURE · 2026 للقدرات التركيبي الإغلاق :HyperAbility
HyperAbility: الإغلاق التركيبي للقدرات
Composed capability closure
HyperAbility تجيب عن مشكلة أن task معقدة لا تنفذها مهارة واحدة. لكنها ليست persona agent؛ هي graph من capabilities ذات شروط إغلاق.
Deliver Production« قد يتكون من discovery، architecture check، edit، tests، security review، WorldEffect verification. مثال »Feature«
.Native/Hermes/Codex/Claude إلى عقده route يمكن graph نفس
Discover → Model → Edit → Test → Verify → Close
HyperCapability Closure Graph - HCCG
نقترح أن كل effectful edge يجب أن يؤدي إلى rollback/verifier أو justification. إذا كان graph يسمح »production write »ثم ينتقل مباشرة إلى
done، فهو غير مغلق. هذا يختلف عن DAG workflow العادي: HCCG يحمل semantics للقدرة والسلطة والدليل.
Dynamic Ability
لا نحتاج تخزين كل composition. planner Mission يمكن أن يبني DynamicHyperAbility من skills متوافقة. إذا تكرر pattern وحقق transfer،
يتحول إلى candidate. هذا يمنع explosion في registry.
- AbilityValue ≈ TransferGain + CompletionGain - CoordinationOverhead
RigidityCost
ً من dynamic planner. لذلك الإغلاق النظري لا يدافع عن HyperAbility كفكرة مقدسة؛ يدافع
إذا كانت ability ثابتة تمنع adaptation، فهي أقل ذكاء
عن properties closure/contract سواء كانت composition مخزنة أو مولدة.
Governed Causal Closure · MOUSSAIF AGI 16 / 30
<PARSED TEXT FOR PAGE: 17 / 30>
الإغلاق :HyperLoop MUT · MUG · MOUSSAIFXT · AL RAZZI · THEORETICAL CLOSURE · 2026 الفجوة فتح وإعادة الزمني
HyperLoop: الإغلاق الزمني وإعادة فتح الفجوة
Temporal closure and gap reopening
الاستقلالية الطويلة تخلق مشكلة جديدة: agent يستطيع الاستمرار طويلاً في الاتجاه الخ طأ. لذلك HyperLoop ليس recurrence فقط؛ هو جهاز إغلاق زمني
يحفظ state ويعيد مقارنة المهمة الأصلية بالواقع بعد كل milestone.
↺ OBSERVE → Δ → PLAN → ACT → READBACK → VERIFY
الذاكرة هي[recovery. RemainingGapو checkpoint، stop conditions، watchdog، budgets للوقت/tokens/tools/iterations يملك loop كل
المنطقية للفرق غير المغلق. لا يكفي أن subtask انتهى؛ يجب أن يعاد فتح Mission Original لاكتشاف requirements التي سقطت أثناء

التنفيذ.
Temporal Closure
Mission طويلة لا تكون مغلقة إذا كانت صحتها تعتمد على بقاء process واحد أو window context واحدة. يجب أن تستطيع استئنافاً معقولاً بعد restart:
،tool call كل replay يعني لا الاستئناف .evidence، known failures، critical decisions، current DAG، restore mission،
external effects
.side effects تكرار يمنعان checkpointsو idempotency
Stagnation: إذا بقي RemainingGap ثابتاً بينما ترتفع الموارد، يعلن stagnation loop بدلاً من الاستمرار. MetaSupervisor يستطيع تغيير
/hypothesis
.human decision يطلب أو ،model/harness/context strategy

<PARSED TEXT FOR PAGE: 18 / 30>

Mesh HyperAgent: الإغلاق التنظيمي
Bounded agent organization
في هذا الإطار الوكيل ليس »شخصية« بل executor bounded يُركّب من scope، Loops/Skills/HyperAbilities، model
/knowledge، scope goal
أو معرفة لتكرار حجة agent identity يصبح أن تمنع الصياغة هذه .authority ceilingو evidence، memory، tools، host policy
contract، budget
سلطة.
أنماط mesh المسموحة
.Planner → Executor → Independent Verifier
.Supervisor → bounded workers مع delegation contract
.Researcher → Synthesizer → Falsifier
.خارجي evaluator مع QMCD candidate agents
.specialist Retained طويل الأفق عندما قيمة retention أعلى من rehydration context
Agent Population Controller
Spawn ⇔ ExpectedParallelOrInformationGain > Coordination + Context + Cost
يستورد لا parent. stop condition، output schema، budget، authority، input handles، goal يحتاج child كل .agent count
زيادة ليس الهدف
.trajectory كاملة إلا عند الحاجة؛ يستقبل RemainingGaps/Artifacts/Evidence/Result. هذا يحافظ على closure context
الإغلاق التنظيمي
يُغلق delegation عندما يعرف النظام من يملك كل node، ما authority الموروثة فعلاً، كيف يعود الناتج، ومن يقرر agent harness
.closure لا External
.nested chaos دون من federated intelligence تسمح القاعدة هذه .global Mission state يملك
سيادة Moussaif: harness/agent قد يكون أقوى محلياً، لكن لا يملك حق إعادة تعريف goal أو criteria acceptance إلا عبر Authority/
Command مسجل.
•
•
•
•
•

<PARSED TEXT FOR PAGE: 19 / 30>

MOUSSAIXT: سيادة

MOUSSAIXT: سيادة المهمة فوق substrate
Cross-environment execution sovereignty
MOUSSAIXT هو طبقة الإغلاق عبر البيئات. إذا كان النظام يعتمد على semantics خاصة بـHermes أو Codex أو Claude، فإنه لا يملك closure
مستقلة؛ يصبح MOUSSAIXT .plugin framework يفصل Contract Mission عن HostAdapter بحيث يمكن للـrouter اختيار substrate وفق
capability/quality/cost/privacy/history.
Mission Node → MoussaifXT Router → Host Adapter → Execution
المورد ما يملكه ما لا يملكه
canonical mission state reasoning/generation Model
global authority local orchestration/tools Harness
semantic success filesystem/process/browser Host
مكتشفة غير provider قدرة يفترض لا goal/state/governance/evidence Moussaif
Cross-Harness Sovereign Routing
Node D عبر Claude verifier. تستخدم أن يمكن واحدة Node A عبر native runtime، Node B عبر Hermes، Node C عبر Codexو، Mission
لكن كل output يعود StructuredObservation إلى plane Evidence/Event. لا يوجد »نجاح Codex «أو »نجاح Hermes «منفصل عن WorldEffect
mission.
شرط portability
نقيس القدرة نفسها عبر hosts مختلفة. إذا لم يمكن تنفيذ HyperAbility إلا لأنها تعتمد على hidden prompt أو path خاص بـprovider، فهي capability
غير مكتملة أو specific-host ويجب وسمها كذلك.

<PARSED TEXT FOR PAGE: 20 / 30>

الفعل: Use
الفعل: Use Computer/Tool كتحول سببي
Action as governed world transition
الفعل هو المكان الذي تتحول فيه cognition إلى transition world، ولذلك يجب أن يكون أكثر طبقات النظام انضباطاً. Action Carrying-Evidence
بعد لا ،التنفيذ قبل verification methodو evidence requirement، reversibility، expected delta، authority يحمل effectful intent كل أن يعني
وقوع الضرر.
{ ActionIntent
capability, target,
expected_delta,
authority,
reversibility,
idempotency,
evidence_required,
verifier
}
Action hierarchy
للـbrowser/computer نفضّل السطح الأكثر تحديداً وأقل غموضاً:
API → CLI/Data → DOM → Accessibility → Vision/GUI

أفضل فهو UI Automation state أو DOM أو database قراءة أمكن إذا .ًأخيراً خيارا screenshot-only تجعل StateAct/OSWorld-style lessons
من inference بصري لكل click. لكن model visual يبقى ضرورياً حين لا توجد structured interface.
Computer closure
نقول success. نعيد observe active window/DOM/accessibility/screenshot/state ونقارن expected effect. popup، focus loss، لا click بعد
recovery/human gate. مع state transitions تصبح CAPTCHA أو stale page
قاعدة الأمان: action computer native-provider لا يتجاوز ToolGateway. النموذج يقترح؛ Moussaif يقرر السلطة وي ُنفذ أو يرفض ثم يتحقق.

<PARSED TEXT FOR PAGE: 21 / 30>

WorldEffect: للنجاح النهائي المعنى Transaction
Transaction WorldEffect: المعنى النهائي للنجاح
Verified world-state success
فنّعر .السببي الإغلاق نقطة هو WorldEffect transaction فيها before state، intended delta، actions، observed after، evidence وverification. هذا يطبق الفكرة الأساسية التي أكدت عليها stateful benchmarks: code return أو message final ليسا correctness state.
WorldEffectTxn {
before,
intended_delta,
action_refs,
observed_after,
observed_delta,
evidence_refs,
verification,
residual_gaps
}
Success = AcceptanceCriteria ∧ VerifiedObservedDelta ∧ ¬CriticalGap ∧ GovernanceOK
درجات النتيجة
الحالة التعريف
mandatory acceptance + verified effect SUCCEEDED.
PARTIAL تقدم مثبت لكن criteria كاملة غير محققة.
dependency/authority. يحتاج معروف gap BLOCKED
FAILED لا أثر مقبول أو regression غير قابل للإصلاح.
before state. إلى أعيدت effect ROLLED_BACK
لماذا هذه نقطة إغلاق نظري؟
لأنها توحّد coding والبحث والautomation. في coding، الأثر state repo/build/test. في research، الأثر set claim صار أوضح مع /evidence
.واحدة structure لكن verifiers تختلف .form/database state، browser task في falsifiers.
Rate Closure False يجب أن يصبح metric رئيسياً: نسبة المهمات التي أعلن النظام نجاحها ثم كشف verifier أو probe state أن شروط القبول لم تتحقق.

<PARSED TEXT FOR PAGE: 22 / 30>

Epistemic والمحاكم الحقيقة إغلاق:

Ledger
Ledger Epistemic والمحاكم: إغلاق الحقيقة
Evidence, claims and independent verification
Ledger Evidence يقول ما الذي لوحظ. Ledger Epistemic يضيف ما الذي يعتقد النظام أنه صحيح ولماذا. كل Claim يحمل
confidence، status،
counterevidence/evidence، scope، freshness وdependencies. لا يزيد status لأن عدة agents كررته؛ الرفع يحتاج evidence أو
experiment.
المعنى Claim status
UNKNOWN لا معرفة كافية
HYPOTHESIS تفسير قابل للاختبار
SUPPORTED أدلة جزئية
VERIFIED ضمن scope ومعيار محدد
متعارض evidence CONTESTED
FALSIFIED اختبار داعم للنفي ضمن scope
state/evidence صلاحية انتهت STALE
Courts
Court Evidence يفحص كفاية الدليل؛ Court Falsification يبحث عن حالة مضادة؛ Court Regression يحمي السلوك السابق؛ Court
Architecture
يحمي invariants؛ Court Security يفحص authority/effect؛ Court Closure يفصل في done؛ Court Generalization يحكم نقل
الـ خبرة إلى
capability.
الاستقلال لا يعني دائماً model آخر، يمكن أن يكون test deterministic أو analyzer static أو predicate database أقوى من verifier
LLM. المحكمة
تختار أرخص source evidence قادر على التفريق.
الإغلاق المعرفي: »لا أعرف« و»متنازع« حالات صحيحة. إجبار claim على false/true يخلق closure false معرفياً مثلما يخلق closure false
message final
تنفيذياً .

<PARSED TEXT FOR PAGE: 23 / 30>

الذاكرة والـخبرة: الإغلاق عبر الزمن
Memory as validated experience
الذاكرة لا تُفهم كـstore واحد. نميز: WorkingMemory للحالة قصيرة الأجل؛ MissionMemory للقرارات والتقدم؛ ProjectMemory
للحقائق المحلية
المستمرة؛ ProceduralMemory للمهارات؛ ErrorMemory لتواقيع الفشل؛ HyperKnowledge للمعرفة الموثقة؛ ExperientialMemory
للمسارات المجردة.
هذا الفصل يمنع logs raw من التلوث كحقائق دائمة.
Experience Compiler
Trajectory → Diagnosis → Experience → Candidate → Evaluation
أو KnowledgePatch توليد يمكن ذلك من .court result، failure/recovery، observed effect، action، decision، context،
cost :إلى ضغطُ ي المسار
SkillPatch أو RoutingPolicyCandidate. لكن الـام يبقى trace للـaudit.
Negative Knowledge
نحتفظ أيضاً بما ثبت فشله: configuration، approach، misuse tool، pattern context. لكن scope ضروري؛ failure في repo واحد
لا يصبح
prohibition universal. ErrorMemory تربط signature بالسبب والاسترداد الناجح.

شرط transfer
الذاكرة تعتبر ذات قيمة فقط إذا حسنت مهمة لاحقة measurable: success أسرع، context أقل، recovery أفضل، أو خطأ متجنب.
otherwise هي
archive.
TransferGain = Performance_with_memory - Performance_without_memory
هذه المعادلة تسمح بفصل »النظام يتذكر« عن »التذكر مفيد«. وهي أحد أهم اختبارات HyperSkill/HyperKnowledge.

<PARSED TEXT FOR PAGE: 24 / 30>

التطور الذاتي: إغلاق التغيير على نفسه
Governed self-evolution
:لاصة ال. Harness Tampering ومخاطر Hermes self-evolution، GEPA، NVIDIA AVO، Prime Continual Harness،
AlphaEvolve هنا تلتقي
improvement-self الحقيقي لا يعني أن agent يعدّل prompt ثم يحفظه. يجب أن توجد candidate + lineage + evaluation executable + independent promotion gate.
Trace → Candidate → Shadow → Held-out Eval → Court → Canary → Promote/Rollback
HarnessGenome
active configuration: context policy، routing policy، model/harness selection، skills/abilities/loops، verification، depth يمثل
recovery. كل genome production له version وparent وscores benchmark. candidate لا يملك حق تعديل evaluator الذي سيحكم عليه.
Agentic Variation Operator
variation. قد يستقبل AVO- best candidate، lineage، failure evidence، objective وremaining gaps ثم يولد hypothesis للتغيير
style operator
فقط. random mutation لا، new skill أو routing change أو context change أو simplification تكون
Evolution closure
self-evolution يجعل هذا rollback path. مع canary ويمر tamper/security regression، دون held-out tasks على gain يثبت عندما مغلق التغيير
جزءاً من engineering production لا ادعاء ذاتي.
asymmetry Constitutional: النظام قد يحسن capability لكنه لا يمنح نفسه authority. أي candidate تمس Authority/Constitution تقع في أعلى risk class وتتطلب حكماً مستقلاً وربما موافقة بشرية.

<PARSED TEXT FOR PAGE: 25 / 30>

مبدأ أضعف حلقة: حدود القوة الكلية
Weakest-link principle and closure entropy
تسمح نظرية الإغلاق بصياغة فرضية مركزية: في mission ذات path critical، قدرة النظام الفعلية لا تتحدد فقط بأقوى component، بل قد تكون
self- مفقود، verifier تعوض لا ممتازة unreliable tool; memory يعوض لا ممتاز reasoning نموذج. bounded by the weakest closure layer
evolution ممتاز لا يعوض benchmark corrupted.
EffectiveCapability ≤ f(min CriticalClosureLayer)
هذه ليست formal theorem مثبتة، بل hypothesis تصميمية. ويمكن اختبارها بتعمد خفض dimension واحدة بينما تثبت البقية. مثال: نفس model/
harness مع readback معطل؛ نراقب closure false. أو نفس system مع disabled memory؛ نراقب transfer. أو نفس stack tool بلا

authority
gate، نراقب policy violations.
Agentic Closure Entropy
نقترح مقياساً مفاهيمياً للفوضى المتبقية في المهمة: عدد/شدة claims، context stale، effects unverified، agents conflicting، gaps، unresolved. يمكن
تسمية هذا Entropy Closure بشرط ألا نخلطه بentropy في نظرية المعلومات دون اشتقاق. وظيفته operational: عندما تنخفض الفوضى المنظمة يصبح
closure أقرب.
H_close ∝ GapDiversity + ClaimConflict + StateUncertainty + UnverifiedEffects
الهدف ليس تقليل كل uncertainty؛ research قد يحتاج إبقاء فرضيات. الهدف تحويل uncertainty غير المعروفة إلى objects معلومة الحالة. لذا system
mature يستطيع أن يقول »هذه ثلاث نقاط unresolved »بدلاً من narrative متماسك زائف.
تنبؤ قابل للهدم: إذا كانت الأنظمة ذات طبقات closure أقوى لا تقلل variance k^pass/success-false مقارنةً ببساطة أقل تحت budget مماثل، فإن الفرضية
link-weakest لا تبرر التعقيد.

<PARSED TEXT FOR PAGE: 26 / 30>

التعميم والنقل: متى تصبح القدرة عامة؟ :والنقل التعميم متى تصبح القدرة عامة؟
Generalization and transfer
engineering oriented-AGI لا يُقاس بعدد capabilities بل بقدرتها على النقل. نعرّف Criterion Transfer Capability: capability تتقدم من
specific-project إلى generalizable عندما تنجح عبر contexts مختلفة مع بقاء العقد الأساسي صالحاً، وبلا dependence hidden على /provider/prompt
repo خاص.
الرتبة الاختبار
Local نجاح في مهمة/بيئة واحدة
Repeated نجاح متكرر في نفس domain
held-out repos Cross-project أو users
مختلفة models عبر capability نفس Cross-model
Cross-harness عبر Native/Hermes/Codex...
domain-Cross آلية تنتقل مع adaptation معقول
Generalization integrity
UI stateو HTTP readback عام، مفهوم« verify external effect»ً مثلا .implementation-specific adapter عن portable contract فصل يجب
verifier تطبيقان. HyperAbility العامة يجب أن تعلن points extension بدلاً من إخفاء assumptions.
من التخصص إلى العمومية
العمومية ليست أن agent واحد ينفذ كل شيء بنفس loop. العمومية قد تكون قدرة النظام على تركيب harnesses/capabilities/knowledge جديدة تحت
نفس laws closure. هذا يجعل compositionality أقرب معيار لنا من »list skill universal«.
Generalization ≈ Reusable Contracts + Adaptive Routing + Transfer Evidence
هذه الصياغة تربط MOUSSAIXT/Abilities/HyperSkills بالبرنامج العلمي: إذا كان النقل ينهار عند تبديل model، فالمعرفة ليست level-OS بعد؛ إذا كان
النقل ينجح، نقترب من claim أقوى.

<PARSED TEXT FOR PAGE: 27 / 30>

القضايا النظرية العشر القابلة للاختبار

Ten falsifiable propositions

القضية اختبار الهدم

Context virtualization P1 يحسن EVCF vs long-context/RAG/RLM fixed-model .long-horizon efficiency.

Active Difference P2 يقلل MCD/Δ routing vs generic planner .compute wasted.

Governing Question P3 عيّسر time/cost-to-decision .resolution.

HyperSkill contracts P4 تزيد contract skill vs prompt procedure .transfer.

HyperAbility closure P5 تخفض HCCG vs dynamic DAG only .integration failures.

Gap reopening P6 يخفض false closure on hidden requirements .premature stop.

WorldEffect readback P7 يخفض message/tool-code vs state verifier .false success.

Cross-harness sovereign routing P8 يحسن single harness vs federation matched budget .robustness.

Experience compilation P9 يحسن transfer gain held-out .task-2 performance.

Governed evolution P10 يرفع performance بلا shadow/canary + tamper audit .tamper.

القضايا P-10P1 هي »قلب الورقة« أكثر من المصطلحات. إذا فشل عدد منها، يجب تعديل النظرية. مثلاً إذا 4P تفشل باستمرار، يمكن الاحتفاظ بـHyperSkill

كـconvenience packaging لا claim ذكائي. إذا 7P تنجح بقوة، WorldEffect يصبح empirical contribution واضحاً.

تحويل النظرية إلى science

baseline، fixed backbone، matched budget، multiple runs، intervals، task stratification، failure taxonomy، raw. traces :قضية لكل

binary success، نضيف partial progress، recovery، constraint violations، time-to-detect-failure، context cost. في نختزلُي لا الأداء

قاعدة الهدم العليا: النظرية لا تُحفظ بإضافة exceptions بعد كل failure. إذا لم تنتج predictions مميزة عن architectures أبسط، فهي إطار وصفي لا نظرية
تشغيلية.

<PARSED TEXT FOR PAGE: 28 / 30>

برنامج الإثبات: من 4E3/E إلى 5E

Experimental closure program

الإغلاق النظري يحدد الآن برنامج الإثبات بدل إنتاج مزيد من corpus. نقترح ست حزم benchmark مترابطة، تعيد استعمال برنامج I Paper لكن تربطه
مباشرة بالقضايا P-10P1.

Track الهدف أهم المقاييس

success, resident tokens, latency EVCF/MCAS vs baselines Context

tests, regression, restart, cost build/fix/refactor real repos SWE

pass^k, false closure, violations stateful user-policy tasks Tool/State

action success, recovery, state correctness long browser/desktop workflows Computer

held-out gain, tamper incidence skills/routing/context candidates Evolution

verified completion, cost, failover Native vs Hermes/Prime/Codex/Claude Federation

Ablation ladder

/A0 → +Context → +EVCF → +HyperKB/PEK → +MCD → +HyperCapabilities → +Courts
WorldEffect → +Evolution → +Federation

كل layer يجب أن يثبت contribution marginal. وهذا يمنع مشكلة كبيرة في المشاريع الطموحة: أن تتحول كثافة architecture إلى immunization ضد
القياس. إذا layer لا تضيف قيمة قابلة للقياس، نحذفها أو نخفض رتبتها.

Production canary
mission لكل. browser task، research synthesis، artifact، بناء code، إصلاح :المخاطر منخفضة real projects نختار ،الاصطناعية benchmarks بعد
self- لا evidence comparative repeated :E5 attainable يصبح هكذا. cost و، evidence، effects، MissionContract، trace، court result ننشر
report.
Governed Causal Closure · MOUSSAIF AGI 28 / 30
<PARSED TEXT FOR PAGE: 29 / 30>
CLOSURE THEORETICAL · RAZZI AL · MOUSSAIFXT · MUG · 2026 ؟ عقد الإغلاق النظري: ماذا نثبت وماذا نترك مفتوحاً MUT ·
عقد الإغلاق النظري: ماذا نثبت وماذا نترك مفتوحاً ؟
Closure contract, limitations and open variables
الإغلاق الحقيقي يجب أن يصرّح بما لن يدعيه. هذه الورقة لا تثبت أن MUG/MUT نظرية عامة للذكاء، ولا أن MCD أفضل من routers أخرى، ولا أن
EVCF يتفوق على RLM، ولا أن HyperSkills شرط للAGI، ولا أن RAZZI AL أقوى من Claude/Codex/Hermes/Prime. كل ذلك ينتظر
evidence.
ما نغلقه
الأساسي القاموس: Goal, Governing Question, Active Difference, Knowledge, Capability, Loop, Agent, Action, Evidence, WorldEffect, Memory/Evolution.
harness/runtime للـ RAZZI AL، substrates عبر للتنفيذ MOUSSAIXT، للحاكمية MUG، للنظرية/lineage، الملكية: MUT.
النجاح معنى: state/evidence closure لا output.
التطور معنى: candidate → independent evaluation → promotion، لا self-edit مباشر.
السياق معنى: virtual address space + resident model working set.
ما نتركه مفتوحاً
الصيغ الدقيقة لـMCD weights وrouting learned.
representation JMCAS/EVCF أفضل.
مقدار structure الأفضل في HyperAbility مقابل dynamic planning.
أفضل courts ومتى تفعل.
حدود self-evolution الآمنة.
هل architecture integrated تتفوق فعلاً على harnesses أبسط.
Contract Closure Theoretical: أي concept جديد بعد هذه الورقة يجب أن)أ(يحل failure لا تمثله primitives الحالية، أو)ب(يكون specialization/
implementation واضحاً. وإلا يدخل كـcandidate note لا theory core.
بهذا يكون »الإغلاق« أداة ضد التضخم النظري. قوة البرنامج القادم تأتي من benchmark، لا من إضافة أسماء. وكل فشل محترم يضغط architecture نحو
نسخة أصغر وأصلب.

-
-
-
-
-
-
-
-
-

•
•

<PARSED TEXT FOR PAGE: 30 / 30>

الإغلاق :اتمةا*ال
ال*ا*اتمة: الإغلاق الذي يفتح مرحلة الإثبات
Conclusion and bibliography
ال*ا*لاصة: تقترح هذه الورقة أن الوحدة الأساسية للذكاء الوكيلي ليست call model ولا persona agent، بل حلقة إغلاق حاكمية سببية مستمرة.
MUT يغلق المعنى
النظري؛ MUG يغلق السلطة والمراتب؛ QMCD/MCD يوجهان compute والبدائل؛ MCAS/EVCF يحفظان السياق خارج حدود النموذج؛
PEK يmaterialize
set working، Loop/Ability/Skill/HyperKnowledge تحول المعرفة إلى قدرة؛ MOUSSAIXT يحافظ على السيادة عبر hosts؛
ToolGateway يحكم الفعل؛
Courts/Evidence/WorldEffect يغلقون الواقع؛ Evolution/Experience يحولون الأثر إلى قدرة مستقبلية.
→ Goal → Δ → Q → Knowledge/Context → Capability → Action → World → Evidence
Closure → Experience → Evolution
إذا أثبتت ablations مساهمة سببية لهذه السلسلة، يصبح MOUSSAIF/RAZZI AL برنامج research systems متميزاً. وإذا فشلت طبقات بعينها، فإن الإغلاق النظري
يفرض تقليصها. هذه القابلية للهدم هي المقصودة من »الإغلاق«: تثبيت الأسئلة والعقود بحيث يصبح المستقبل اختباراً لا زخرفة.
المراجع ال*ا*ارجية - المجموعة الموروثة من I Paper
[R1 Karten et al. (2026), Prime Agent: A Self-Improving RLM Harness, arXiv:2608.23552.
[R2, Chen et al. (2026), Agentic Variation Operators arXiv:2603.24517.
[R3, Lumer et al. (2026), Recursive Agent Harnesses arXiv:2606.13643.
[R4, Zhang, Kraska, Khattab (2025), Recursive Language Models arXiv:2512.24601.
[R5 Yu et al. (2026), Recuris: Recursive Experiential-Working Memory Evolution, arXiv:2608.24876.
[R6. Agrawal et al. (2026), GEPA, ICLR 2026
[R7 Lehmann et al. (2026), Tycho: Active Abstraction with Programmatic World Models.
[R8. Yuan et al. (2026), OSWorld2.0
[R9. Yang et al. (2026), StateAct
[R10- Wang et al. (2026), Auditing Harness Tampering in Self Improving Agents.
[R11 Luo et al. (2026), From Storage to Experience: Survey of LLM Agent Memory, ACL Findings.
[R12. Xu et al. (2025), A-MEM: Agentic Memory for LLM Agents
[R13. Novikov et al. (2025), AlphaEvolve
[R14. Fourney et al. (2024), Magentic-One
[R15. Schmidgall et al. (2025), Agent Laboratory
[R16. Lu et al. (2025), ToolSandbox, NAACL Findings
[R17. Yao et al. (2025), τ-bench, ICLR

]Zhang et al. (2024), Agent-SafetyBench]R18.
]Pan et al. (2025), SWE-Gym, ICML]R19.
]Yang et al. (2024), SWE-agent, NeurIPS]R20.
]Wang et al. (2025), OpenHands, ICLR]R21.
]Wang et al. (2024), CodeAct, ICML]R22.
]Sumers et al. (2024), CoALA, TMLR]R23.
]Mialon et al. (2024), GAIA, ICLR]R24.
]Xu et al. (2025), TheAgentCompany, NeurIPS]R25.

مراسي المشروع

AL RAZZI / MOUSSAIFXT - 25-Paper Frontier Agentic Harness Research, Paper I, 2026.]M2[MUT/MUG &]Applied Theory]M1

New MOUSSAIF AGI Technologies, Paper II, 2026.]M3[MOUSSAIF AGI OS / MOUSSAIXT Deep Engineering. Architecture, 2026

,MOUSSAIF Unified AGI Research Book, 2026.]M5[AGI-Oriented Knowledge Base as Cognitive OS-]M6[ParallaxWorld]M4[.2026

MCD Full AGI OS, 2026.