HEADSOFT RESEARCH / WORKING PAPER 0.1

Equal Evidence, Different Outcomes: A Process-Isolated Evaluation of Governed External Reasoning Around Small Language Models

Richard T. Elmore | HeadSoft Research | July 14, 2026 | doi:10.5281/zenodo.21367631

> The numerical claims are tied to a machine-verifiable evidence ledger. This is a working technical report, not an official benchmark submission or independent replication.

AMOEBA COMMIT	MANUSCRIPT SHA-256
b669affdb769	116731c67dcb112e...

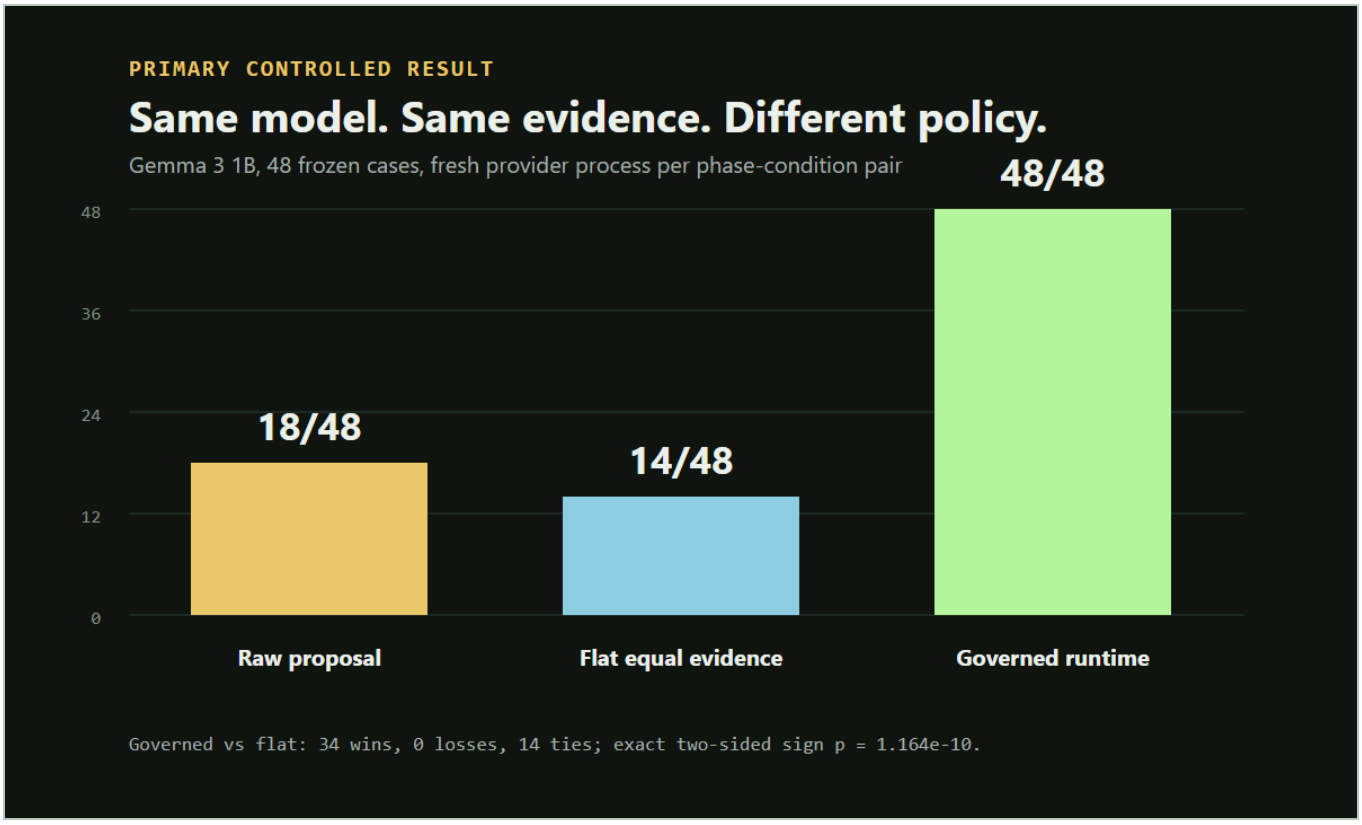

Figure 1. Primary process-isolated target result. The reverse-order replay reproduced the same scores exactly and is not counted as an independent sample.

Abstract

Language-model systems are often improved with retrieval, memory, retries, tools, and verification, but measured gains can be difficult to attribute when conditions differ in evidence, compute, persistent state, or evaluator access. We present Amoeba Core v2, an experimental external intelligence substrate in which a language model supplies bounded proposals while deterministic records, selectors, evidence authority, proof obligations, and route policies govern how those proposals may affect an answer. The central question is not whether more context helps, but whether governed processing changes outcomes when the model, evidence, and work are held fixed.

Persistent records survive individual model calls and distinguish provenance, scoped trust, claim confidence, and execution authority. This external memory does not enlarge the model's finite context window; it lets policy select which durable records should enter a later bounded context or control a deterministic substrate operation.

Our primary experiment froze a generated 48-case opaque-signature task family before one target execution with Gemma 3 1B. We compared a raw proposal, a flat prompt given acquired evidence, and a governed route given exactly the same evidence and matched work. Every phase-condition pair ran in a fresh provider process, and a reverse-order phase tested replay and order effects. In both phases, raw scored 18/48, flat equal evidence scored 14/48, and governed scored 48/48. Governed versus flat

produced 34 paired wins, 0 losses, and 14 ties (`p=1.164e-10`, exact two-sided sign test) per phase. Raw and governed began from identical model proposals on all 48 cases; flat and governed matched on evidence identity, calls, experiments, and normalized work. The replay phase was exact but is not counted as an independent second sample. A separately committed internal post-hoc verifier recomputed all persisted semantics without rerunning the target or calling the model.

Secondary results show both capability and present limits. On a local, closed-book, nonofficial HumanEval run, one raw Qwen 2.5 Coder 14B completion per task passed 129/164 and one Amoeba primary-route completion per task passed 164/164 without post-test candidate selection. However, the methods were developed while diagnosing HumanEval failures. On a synthetic 72-case transfer set, top-4 surface retrieval tied Amoeba at 72/72. On a project-untouched 16-case family holdout, a frozen selected policy reached 14/16 versus 12/16 for the static policy, a non-significant paired result (`p=0.5`). These results support a narrow causal claim: governed deterministic evidence processing can change a fixed model's outcomes beyond flat evidence injection in a controlled family. They do not yet establish broad learning transfer, general superiority over retrieval, official benchmark standing, or autonomous scientific reasoning.

---

1. Introduction

Large language models can propose useful answers, plans, programs, and tool calls, but they do not by themselves provide durable state isolation, typed authority, provenance-complete evidence selection, or deterministic proof policy. Agent systems add memory and feedback around models, yet the resulting score improvement is often compatible with several explanations: more useful text was retrieved, more candidates were sampled, tests acted as an answer oracle, persistent provider state leaked across conditions, or a benchmark was used to author the intervention being evaluated.

Amoeba Core v2 is an experiment in moving operational authority out of the language model and into a persistent, inspectable substrate. The model is treated as one proposal-producing component. Evidence, methods, policies, tools, costs, and proof results are typed records with provenance. A selector may admit a route only in a declared operating context, and a verifier may reject or localize a proposal without asking the model to narrate a more persuasive answer. This architecture does not make stochastic generation deterministic. It makes the state transitions surrounding generation explicit, replayable, and subject to independent checks.

The empirical challenge is to distinguish this substrate behavior from ordinary context injection. Our primary study therefore compares two evidence-bearing conditions. The flat condition gives a weak local model the exact authorized observations in prose. The governed condition starts from a bounded model proposal and processes those same observations through frozen selector, verifier, localization, and proof machinery. Evidence identity, model calls, experiment calls, and normalized work are matched. Fresh provider processes and reverse condition order address state carryover and order effects.

This paper makes four contributions:

1. It describes a substrate-governed architecture that externalizes selected task state and learned methods into durable active memory while separating proposal, evidence authority, selection, verification, and

promotion. Later behavior can therefore change without changing model weights or replaying the complete interaction history into every model call.
2. It reports a preregistered, process-isolated equal-evidence experiment in which governed processing corrected 30 initially wrong model proposals and reached 48/48 while flat same-evidence prompting reached 14/48.
3. It provides a machine-verifiable evidence ledger linking every quantitative statement to persisted artifacts, SHA-256 digests, commits, and explicit exclusions.
4. It reports counterevidence prominently: a wide retrieval baseline tied Amoeba on a synthetic transfer set, and a project-untouched transfer result was small and non-significant.

The paper does not claim that the individual ingredients are new. Retrieval, external memory, iterative feedback, tool use, programmatic reasoning, and test based selection all have substantial prior art. The contribution evaluated here is the combination of typed governance and causal instrumentation, together with evidence that this combination can be outcome-relevant under matched evidence and work.

The evidence classes are kept separate throughout:

Ledger entry	Evidence class	Role in this paper
E1	Preregistered live, process-isolated causal target	Primary result
E2	Recognized dataset, local nonofficial run	Capability demonstration
E3-E4	Held-out comparator and transfer studies	Retrieval and generalization boundary
E5, E7-E8	Generated component batteries	Mechanism attribution
E6	Post-hoc policy derived from E1	Operational consequence, not a new sample
E9-E11	Small-n live and public-source studies	Proof of concept

2. Related Work

2.1 Retrieval and external memory

Retrieval-augmented generation combines parametric model memory with an external nonparametric store Lewis et al., 2020. MemGPT applies operating-system ideas to context management and long-running memory Packer et al., 2023. These systems establish that information outside model weights can improve or extend language-model behavior. Amoeba shares that premise, but its present research

question is narrower: after evidence has already been acquired, can typed authority and deterministic processing outperform presenting the same evidence as flat text?

2.2 Feedback, reflection, and skill libraries

Reflexion stores linguistic feedback in episodic memory without updating model weights Shinn et al., 2023. Self-Refine iterates model-generated feedback and revision Madaan et al., 2023. Voyager accumulates an executable skill library and uses environment feedback to improve behavior over time Wang et al., 2023. Amoeba similarly preserves methods and failure evidence, but distinguishes proposal-only text from source-backed or proof-backed authority and records the causal path by which a method affected selection.

2.3 Reasoning, acting, and external execution

ReAct interleaves model reasoning and environment actions Yao et al., 2023, while Tree of Thoughts searches among multiple reasoning paths Yao et al., 2023. Program-Aided Language Models delegate exact computation to an interpreter Gao et al., 2023, and Toolformer learns when to call external tools Schick et al., 2023. These works motivate the division of labor between a semantic proposer and deterministic executors. Amoeba goes further in treating tool and model outputs as evidence with explicit authority, cost, provenance, and promotion state.

2.4 Compiled language-model programs and code verification

DSPy represents language-model pipelines as declarative modules that can be compiled against a metric Khattab et al., 2024. CodeT selects generated programs using generated tests and execution agreement Chen et al., 2022. HumanEval introduced a functional-correctness benchmark and pass@k evaluation for code generation Chen et al., 2021. These systems make clear that orchestration and verification can improve outcomes without weight updates. Accordingly, this paper separates first-route behavior from post-test candidate selection and does not label best-of-n verification as learning.

---

3. Amoeba Core v2

3.1 Model as a bounded proposer

The substrate does not assume that a language model is a trusted reasoner or database. A model call is an organelle invocation with a model identity, provider, ABI, seed capability, role, cost, and provenance record. Its output is proposal material until a policy grants stronger authority. The same interface can in principle wrap calculators, search tools, sensors, code executors, or human operators, although this paper evaluates local language models and deterministic executors.

3.2 Typed records and active memory

Durable substrate state contains typed claims, observations, methods, anti-methods, policies, proof results, and behavior-change traces. Records can carry source identity, timestamps, contamination or taint state, scientific status, trust scope, and links to dependencies. Retrieval identifies candidates; selector gates decide which candidates may enter active context or control execution. This distinction is important because availability does not imply authority.

At runtime, Amoeba can assemble a bounded context packet from selected records instead of replaying the full conversation or learning history. A validated method can also control a deterministic substrate route without being converted back into prose for the model. Learning in this architecture therefore means changing durable, inspectable substrate state and its selection policy rather than modifying opaque model weights. Whether such learned state transfers to untouched task distributions remains a separate empirical question.

3.3 Selector, verifier, localization, and proof

For the primary study, the relevant governed path is:

```
model proposal
  -> typed candidate hypothesis
  -> authorized observations
  -> deterministic signature verifier
  -> mismatch localization when needed
  -> proof obligation resolution
  -> final governed selection
```

The verifier owns the admissibility test. A model may nominate a hypothesis, but it cannot promote its own self-report to independent evidence. When a proposal conflicts with observations, localization identifies the failed obligation and the selector evaluates the remaining candidates. The final answer is therefore a substrate state transition, not merely the model's last utterance.

3.4 Knowledge-access lanes and claim authority

Amoeba separates `closed_book`, `open_book_live`, and `prestudy_frozen` evaluation lanes. Reports declare allowed tools, source and contamination policies, and whether official-score language is enabled. In this paper, the HumanEval result is closed-book; the small public-source demonstrations are prestudy-frozen and cannot support closed-book claims.

3.5 Durability, determinism, trust, and scientific-method scope

Durable memory. Amoeba stores evidence, methods, policies, proof results, and behavior-change traces outside model weights and request context. Records can survive individual calls and sessions and can be selected for later work, so useful state need not remain resident in one finite context window. This is external state management, not an enlargement of the model's per-call context. The evidence in this paper demonstrates persisted records, later source-backed reuse, and causal dependence on selected evidence. It does not establish unbounded operating duration, storage scale, or broad untouched learning.

Determinism. Amoeba uses deterministic canonical records, hashes, selector and verifier rules, proof executors, and replay checks where the underlying operation permits them. Language-model generation remains stochastic and can remain provider-dependent; fixed sampling, seeds where supported, and process isolation are experimental controls rather than a universal determinism guarantee. The primary result establishes exact replay for one frozen governed route, not deterministic language modeling in general.

Provenance, trust, confidence, and authority. These are separate substrate dimensions. Provenance records origin and transformation lineage. Trust is a scoped assessment of a source or record given its identity, history, domain, taint or contamination state, alignment, and available proof. Confidence describes current support for a claim. Authority determines which downstream uses policy permits. A model's or source's self-reported confidence does not grant itself authority, and a provenance link authenticates lineage rather than truth. The present experiments exercise authority and taint controls; they do not establish universally calibrated trust estimates.

Scientific-method orchestration. The implemented bounded gate can require a hypothesis, risky prediction, source observation, internal experiment or proof, confidence update, cost decision, and final selector decision. E9 exercises that sequence in one live, process-isolated session with component controls. This shows that the workflow can be substrate-enforced rather than merely suggested in a prompt. Its sample size does not establish autonomous science, open-domain discovery, or the correctness of arbitrary experiments.

3.6 What remains architectural rather than empirical

The repository also contains typed skill IR, cost scheduling, model-role metadata, evolutionary policy search, provenance-aware replay, and incremental dependency concepts. Some have generated mechanism tests, but they are not all validated on external tasks. They are treated as implementation or research hypotheses unless a result in the evidence ledger says otherwise.

---

4. Research Questions

RQ1. Equal evidence: Can the governed route outperform a flat same-model route when evidence identity, model calls, experiment calls, and normalized work are matched?

RQ2. Causal location: If outcomes differ, does the difference occur before or after the model proposal?

RQ3. Reproducibility: Does the result survive reverse condition order, fresh provider processes, exact semantic replay, and independent artifact verification?

RQ4. Capability: Can substrate routing improve a local model on a recognized programming dataset without post-test candidate selection?

RQ5. Alternative explanations: How much of the measured gain can be matched by retrieval, exact deterministic calculation, or selected retry, and does the method transfer to a project-untouched family?

---

5. Methods

5.1 Evidence discipline

Every result used in the paper is registered in `evidence_ledger.json`. Each entry declares its evidence class, independence group, source documents, artifact hashes, quantitative assertions, supported claim, excluded claims, and official-score status. The verifier recomputes 219 ledger checks without executing a target, benchmark, evaluator, or model.

Replays, post-hoc verifiers, and derived policy analyses share the independence group of their source experiment. They are not pooled as new samples. A benchmark-developed method is reported as a capability demonstration, not as untouched transfer. A selected-candidate result is not reported as pass@1.

5.2 Primary task family

The target contains 48 generated cases. Each case defines four opaque mechanisms and three opaque experiments. Every mechanism has a distinct three-experiment signature generated from two latent bits and parity. The public packet includes opaque aliases (`H001` through `H004`), predicted signatures, and an explicitly ineligible same-proposer self-report. The sealed packet contains the expected mechanism and observations. The public task packet was shown to the model; the sealed expected mechanism was not, and target truth was not supplied as answer material.

The family is intentionally diagnostic. It tests whether a weak model can use or be governed by structured evidence, not whether it possesses world knowledge. Because the verifier and task family were co-designed, a purpose-built deterministic algorithm remains a valid alternative solution.

5.3 Conditions

Each case was evaluated under three conditions:

1. **Raw proposal.** One bounded model proposal without acquired evidence.
2. **Flat equal evidence.** The same model receives the exact authorized observations as flat prompt text.

3. **Governed runtime.** The bounded proposal passes through the frozen typed selector, verifier, localization, and proof route using the same authorized observations.

Flat versus governed is the primary score-bearing comparison. Raw is descriptive because it performs no experiments and therefore has lower work.

Flat and governed were required to match per case on normalized experiment identity, outcome, evidence references, observation cost, model-call count, experiment-call count, evidence work, and total work. Prompt tokens were measured rather than forced equal; flat received more.

5.4 Model and provider isolation

The model was Ollama `gemma3:1b` with temperature 0, top-k 1, top-p 1, a fixed case seed, and a 16-token output limit. The campaign created one fresh provider process for every phase-condition pair, yielding six isolated epochs. Each epoch used a unique process identity and endpoint, acknowledged generation, terminated cleanly, and removed ephemeral roots. Storage paths excluded the system drive.

Phase A ran raw, flat, then governed. Phase B ran governed, flat, then raw. The same 48 cases were used in both phases solely to test order and replay. No phase was selected away.

5.5 Frozen execution and verification

The implementation was committed before target selection. A new target seed, case set, public digest, sealed evaluator commitment, and manifest digest were frozen and pushed before one execution. Passing and failing outcomes were both binding; tuning or rerunning the consumed target was prohibited.

After execution, a separately committed verifier read only the persisted manifest and report. It did not import or call the campaign generator, rerun a condition, call the model or provider, or execute an evaluator. It recomputed matrix completeness, scoring, proposal identity, evidence/work equivalence, paired statistics, replay, authority labels, and provider-process isolation.

5.6 Statistics

The primary paired outcome is governed versus flat correctness on the same 48 cases. We report wins, losses, and ties and use an exact two-sided sign test over discordant pairs. Phase B is a deterministic replay and counterbalance check; its identical statistic is reported but not treated as an independent replication or pooled into `n=96`.

5.7 Secondary studies

The evidence ledger includes five kinds of secondary study:

- a full local HumanEval primary-route comparison;

- a synthetic 72-case method-transfer and retrieval comparison;
- a project-untouched, family-disjoint 16-case transfer replay;
- generated component-ablation batteries for deduction, noisy evidence, and evolutionary policy search;
- small-n process-isolated science and frozen public-source demonstrations.

These studies answer different questions and are not combined into one score.

---

6. Results

6.1 Primary equal-evidence result (E1)

Both phases produced exactly the same score:

Condition	Phase A	Phase B	Model calls/phase	Experiments/phase	Work/p
Raw proposal	18/48	18/48	48	0	48
Flat equal evidence	14/48	14/48	48	96	144
Governed runtime	48/48	48/48	48	96	144

Governed versus flat yielded 34 wins, 0 losses, and 14 ties in each phase. The exact two-sided sign-test probability was `1.164153218269348l4e-10`. The governed route therefore exceeded flat evidence in this designed family despite matched evidence and work.

6.2 The delta occurred after the model proposal

Raw and governed primary proposal aliases and proposal-trace digests matched on 48/48 cases in each phase. Raw was correct on 18 cases, leaving 30 wrong initial proposals. Governed recorded 30 localizations and corrected all 30. This locates the score change after the model proposal, inside deterministic evidence processing and selection.

6.3 More prompt text does not explain the result

Flat and governed matched all 48 cases per phase on normalized evidence and work fields. Flat received 32,477 input tokens per phase while governed received 28,312. Thus the governed advantage was not

caused by a larger prompt-token budget. In this family, putting the evidence into the weak model's context was insufficient for reliable application.

6.4 Replay and isolation

All 144 case-condition rows matched exactly between the two phase orders. All six provider epochs passed process, endpoint, readiness, termination, cleanup, and retained-artifact checks. No raw model text was persisted, and authorized official-evaluator reads were zero.

The separately implemented internal verifier passed all 18 top-level semantic checks, 7 matrix and condition-order checks, 6 scoring and authority checks, and 10 independently recomputed provider-isolation checks. It recovered the same 18/48, 14/48, and 48/48 scores and the same paired comparison without target re-execution.

6.5 HumanEval capability result (E2)

On the 164-task local HumanEval artifact with Qwen 2.5 Coder 14B:

Condition	Passed	Rate
Raw, one completion/task	129/164	78.7%
Amoeba primary route, one completion/task	164/164	100.0%

The delta was 35 tasks or 21.3 percentage points, with no negative deltas. The reported Amoeba score did not use post-test candidate selection or retry. Of the 164 routes, 150 used the normal Amoeba-routed model path, 13 used a proven deterministic substrate renderer, and one used a scoped import-policy renderer.

The method and policy records used by the 14 deterministic or scoped routes were present in durable substrate state before this run. They were selected as the first route and did not require a model-weight update or replay of the earlier blocker-diagnosis sessions. This is direct evidence that persistent substrate state can retain benchmark-developed capability and affect later first-route outcomes. It is not evidence that those methods transfer to an untouched benchmark, because HumanEval failures informed their development.

This result is not an official leaderboard score. Hardened sandbox and official pass@k parity gates remain incomplete. More importantly, the methods were developed while diagnosing this benchmark, so the result demonstrates retained capability and primary-route composition rather than untouched generalization.

6.6 Retrieval explains much of the synthetic transfer result (E3)

On a synthetic 72-case held-out aggregate, raw scored 10/72, flat same evidence 39/72, top-2 surface retrieval 69/72, and typed Amoeba 72/72. Top-4 surface retrieval also reached 72/72. This tie prevents a broad claim that Amoeba's current accuracy exceeds strong retrieval. The measurable distinction on this set is narrower: typed selection reached the ceiling at a smaller retrieval width and added provenance, authority, taint, and proof controls.

6.7 Untouched transfer remains unproven (E4)

On a project-untouched, family-disjoint 16-case holdout, a frozen selected policy reached 14/16 versus 12/16 for a static policy. The paired comparison was two wins, no losses, fourteen ties, with exact `p=0.5`. Budget-matched retrieval reached 13/16 on the primary route; relevant Amoeba primary routes reached 12/16. The selected 14/16 depended on proof-authorized retry or selection.

This is limited evidence. It does not establish that the learned methods or generic pretraining transferred to a new family.

6.8 Mechanism batteries and negative controls (E5, E7, E8)

On a generated answer-hidden 72-case deduction battery, successive conditions scored 2, 10, 20, 24, 36, 60, and 72 correct as obligation, hypothesis, verification, and localization machinery was added. The full route exceeded the strongest verifier-only frontier by 12 wins and no losses (`p=0.00048828125`). No model was called.

On a generated noisy-latent battery, plain exact same-evidence MAP and governed adaptive inference tied at 86/96 clean cases. Governed processing therefore did not improve accuracy over the exact calculator. It did pass all 96 special authority controls versus 16/96 for an ungated adaptive condition. This result supports governance behavior, not mathematical superiority.

On a favorable generated policy-search landscape, archive-guided crossover plus mutation found a complete held-out policy in 12/12 repeats versus 2/12 for the strongest equal-budget comparator. This is evidence that the implemented evolutionary search can be useful in its designed landscape, not that evolution improves recognized benchmarks.

6.9 Small-n live and public-source studies (E9-E11)

One process-isolated Qwen 14B scientific-method session scored 0.3333 in the raw condition and 1.0 under the full substrate gate, with zero control passes across the ablated conditions. Two exact-URL frozen-prestudy studies, one on Python 3.14 t-strings and one on Kubernetes stop signals, each moved the same model from 0/3 raw to 3/3 with frozen source-backed memory while evidence-removed, wrong-study, and unrelated-study controls remained 0/3. These runs show that the live gate and prestudy lanes

operate end to end. Their one-task and three-task sample sizes are not evidence of broad scientific or public-source learning.

---

7. Discussion

7.1 What the primary experiment establishes

The strongest conclusion is causal but narrow. Within one frozen generated family, the same weak model proposal led to different final outcomes because the substrate, rather than the model, owned evidence interpretation and final authority. Retrieval quantity, evidence identity, model calls, experiment calls, normalized work, provider carryover, sampling profile, condition order, and larger prompt size do not explain the governed-versus-flat delta.

This is stronger than showing that memory text helps a model, because flat text contained the same evidence. It is weaker than showing broad learning, because the task and deterministic verifier were designed together and the governed solution resembles an exact symbolic procedure.

7.2 A model can be useful without being authoritative

Gemma 1B did not need to perform the complete inference reliably. It supplied a bounded semantic proposal, after which typed deterministic machinery completed the task. This supports a practical architectural hypothesis: small or eccentric models may be useful as low-cost translators or proposers when the substrate can verify and constrain their contribution. The experiment does not show that every task can be reduced this way.

7.3 Governance and accuracy are distinct outcomes

The retrieval tie and noisy-latent calculator tie are informative. Governance can be valuable even when it does not improve accuracy, because it can preserve source authority, reject forbidden evidence, expose provenance, enforce cost or lane policy, and produce a reproducible trace. Conversely, auditability alone does not justify an accuracy claim. The system should therefore report outcome utility and governance compliance separately.

7.4 Baseline preservation

The flat equal-evidence route scored below raw in the primary family. A derived route-admission analysis did not declare flat harmful because the directional evidence was inconclusive under its preregistered threshold. It instead preserved raw as the default and admitted the governed route only in the exact model, ABI, role, domain, task, proof, budget, skill, and knowledge-lane context where paired value was

demonstrated. This illustrates an important design law: possessing more substrate machinery does not imply that the system should always activate it.

7.5 Durable learning is still an open empirical question

The repository demonstrates durable records, method reuse, and causal dependence on evidence. However, the strongest untouched transfer study has not yet shown significant improvement over its baselines. It is therefore more accurate to say that Amoeba has a working substrate-learning mechanism with bounded causal demonstrations than to say that general learning has been established.

---

8. Threats to Validity

8.1 Designed task family

The primary tasks were generated specifically to exercise opaque hypothesis signatures, evidence authority, and deterministic localization. They are not a natural distribution. A conventional exact algorithm could plausibly solve the same family, so the result validates substrate operation rather than general reasoning superiority.

8.2 Single primary model and local stack

The primary experiment uses one Gemma 3 1B checkpoint through one local Ollama stack. Fresh processes control carryover but do not provide model-family replication. The HumanEval result uses Qwen 2.5 Coder 14B but a different task and intervention regime.

8.3 Replay is not replication

The second phase repeats the same cases under reversed condition order. Exact replay is valuable engineering evidence, but it contributes no new independent task sample. Statistical inference uses one 48-case target.

8.4 HumanEval adaptation and official scoring

HumanEval informed method development, making the 164/164 result unsuitable as untouched evidence. The local harness also lacks completed hardened-sandbox and official pass@k-equivalence gates. No leaderboard or frontier comparison should be inferred.

8.5 Retrieval comparator scope

The top-4 retrieval comparator ties Amoeba on one synthetic set. Other retrievers, embedding models, rerankers, or context budgets were not exhaustively tested. The data neither establishes Amoeba superiority nor proves permanent equivalence to retrieval.

8.6 Internal verification

Artifact hashes and an independently implemented verifier make silent mutation harder and catch several classes of inconsistency. They are still authored and run within the same project. External reproduction and code review remain necessary.

---

9. Claim Boundary

The evidence currently supports this statement:

> Amoeba Core v2 can place deterministic, provenance-bearing evidence and proof authority around a fixed local language model. On one preregistered generated 48-case family, this governed route corrected identical weak-model proposals and reached 48/48 while a flat same-model route with the exact same evidence and matched work reached 14/48, under fresh provider-process isolation and exact replay.

The evidence does not currently support these statements:

- Amoeba generally outperforms RAG or all agent scaffolds.
- Amoeba has demonstrated broad untouched learning transfer.
- Amoeba has an official 100% HumanEval score.
- A 1B model plus Amoeba has frontier-equivalent general coding ability.
- Evolution has improved an external benchmark.
- Amoeba removes the per-call context limit or has demonstrated unbounded long-horizon memory.
- Amoeba makes language-model generation deterministic.
- Amoeba performs autonomous or deterministic science in open domains.

---

10. Reproducibility and Artifacts

The canonical repository root is `H:\amoeba_core_v2`. The paper evidence cutoff is commit `d85ccccaacfd733e8761cff614f3368226d4f890`. The primary target artifact SHA-256 is `2050796f3bf16a72b97fa30b337b472c41f135c9bcc95050b5d5da4944e8b3ef`. The independent verification artifact SHA-256 is `7887e3335bfa9b8fec3c2f1df425c7537407b2a5e51fea3c5a3533597f2f7b56`.

The complete claim map is in `EVIDENCE_LEDGER.md`; machine assertions are in `evidence_ledger.json`.
Run:

```
python tools/verify_paper_evidence.py
  --output reports/paper_evidence_verification_20260714.json
```

The audit command performs no model calls, evaluator reads, or target executions. The consumed primary target must not be rerun.

---

11. Next Experiments

The next paper-grade experiments should be ordered by the alternative explanations they eliminate:

1. Freeze an independent equal-evidence task family whose solution cannot be reduced entirely to the existing signature matcher.
2. Add equal-budget retrieval, retry, verifier, and purpose-built deterministic algorithm baselines to that same target.
3. Replicate with a second model family under the same process-isolation and seed contract.
4. Run a sufficiently powered untouched transfer study in which target outcomes are never used to author or select methods.
5. Complete official sandbox and scoring parity, then run an externally recognized benchmark without benchmark-developed routes.

---

12. Conclusion

Amoeba's current strongest result is not a leaderboard score. It is a causal mechanism result: under matched evidence and work, a deterministic governed route transformed the same weak-model proposal into reliably correct outcomes where flat evidence injection did not. The result survived fresh provider processes, reverse order, exact replay, frozen execution, and independent artifact verification. Secondary studies show that the same architecture can compose a strong local HumanEval route, but also show that retrieval can match it on a synthetic set and that untouched transfer remains unproven. The appropriate conclusion is therefore specific: external substrate governance can be outcome-relevant and auditable; its general learning advantage is the next hypothesis to test, not a result already obtained.

References

1. Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, Sebastian Riedel, Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. 2020.

2. Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, Joseph E. Gonzalez. MemGPT: Towards LLMs as Operating Systems. 2023.

3. Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. Advances in Neural Information Processing Systems. 2023.

4. Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, Peter Clark. Self-Refine: Iterative Refinement with Self-Feedback. Advances in Neural Information Processing Systems. 2023.

5. Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. 2023.

6. Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. International Conference on Learning Representations. 2023.

7. Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, Karthik Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. Advances in Neural Information Processing Systems. 2023.

8. Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, Graham Neubig. PAL: Program-Aided Language Models. 2023.

9. Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, Thomas Scialom. Toolformer: Language Models Can Teach Themselves to Use Tools. Advances in Neural Information Processing Systems. 2023.

10. Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, Christopher Potts. DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines. 2024.

11. Bei Chen, Fengji Zhang, Anh Nguyen, Daoguang Zan, Zeqi Lin, Jian-Guang Lou, Weizhu Chen. CodeT: Code Generation with Generated Tests. 2022.

12. Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter

Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, Wojciech Zaremba. Evaluating Large Language Models Trained on Code. 2021.