Hash-Chained Verbatim Quotation: A Verifiable Provenance Layer for Grounded Retrieval

Mario Gutiérrez
Celiums Solutions LLC
Medellín, Colombia
terrizoaguimor@gmail.com

2026-05-28

Abstract

We study *verifiable provenance* as an addable layer for extractive retrieval systems: a system that answers by emitting byte-identical quotations of stored fragments, over a SHA-256 hash-chained journal, can have any output independently audited back to named, unaltered sources — a guarantee no retrieval-augmented LLM provides by construction, because generation paraphrases its sources. Our worked instance is HYPHAE, a cognitive substrate that composes answers by verbatim quotation joined with a curated lexicon, with no LLM in the cognition path. We evaluate it against 18 LLM-based configurations (vanilla RAG, HyDE plus cross-encoder reranking, oracle context; six generator models — from Anthropic, OpenAI, DeepSeek, two from Meta, and an in-house router — times three modes), plus two controls: a trivial **echo** baseline that emits the retrieved sentence verbatim with no model, and **echo+journal**, echo with the same hash chain. Two findings, both negative for "HYPHAE the system" and positive for the layer. First, on standard correctness and NLI-grounding metrics, echo ties or exceeds HYPHAE on both a 150-query TriviaQA sample and a 34-query composition corpus — measured correctness is a property of verbatim quotation, shared by any echo, not of HYPHAE's composition machinery (our ablations confirm the lexicon, cascade-shape composition, and smoothing do not move the metrics). Second, on a minimal tamper-detection benchmark (four tampering modes × two adversaries, against the real journal), the (verbatim + hash-chain) layer detects and localises 100% of store-only tampering — but so does echo+journal, identically, because the journal stores raw bodies and the realizer is irrelevant to it. Provenance, like correctness, is not HYPHAE-specific; it is an addable layer, and the line that separates systems is journal-vs-no-journal (plain echo and LLM-RAG detect 0%). The guarantee is *tamper-evident*: a chain-aware adversary who rewrites the persisted head defeats the bare chain, but is caught once the head is externally anchored with an Ed25519 signature held outside the store, which we implement and demonstrate — so detection holds against any attacker who does not also compromise the anchor key. The sub-millisecond, CPU-only cost is a corollary of not generating, shared with echo. We argue the durable contribution is the provenance layer and the benchmark critique — the standard correctness benchmark cannot measure either, since a `print` statement saturates it — and that evaluating verifiable generation needs provenance-centric benchmarks the literature does not yet provide. Code, corpora, result envelopes (including both controls), and this paper's source are at https://github.com/terrizoaguimor/hyphae-v2.

1 Introduction

Large language models (LLMs) augmented with retrieval have become the default architecture for grounded factual question answering, conversational memory, and document summarisation.

The dominant pattern — chunk a corpus, embed each chunk, retrieve top-k at query time, and condition an LLM on the retrieved passages — is deployed at substantial cost: a 70-billion-parameter generator loaded into a high-memory GPU instance, billing per token, with per-query latencies in the multiple-second range. Recent extensions of this pattern (HyDE [Gao et al., 2022], RAG-Fusion [Rackauckas, 2024], cross-encoder reranking [Nogueira and Cho, 2019, Reimers and Gurevych, 2019], GraphRAG [Edge et al., 2024]) improve retrieval quality without changing the fundamental architectural decision: the LLM remains the composer.

This paper investigates the alternative. We study the verifiable provenance layer through Hyphae, a cognitive substrate that produces grounded responses without invoking any LLM at inference time and serves as our worked instance of a system carrying the layer. Hyphae achieves grounding not through training but through an architectural commitment: **retrieved memory fragments are quoted verbatim, and the connective tissue between quotations is drawn from a curated lexicon, never generated by a statistical model**. The substrate runs on a single CPU core, holds approximately 50 MB of resident memory, and produces composition outputs in microseconds.

We compare Hyphae against 18 LLM-based system configurations spanning vanilla retrieval-augmented generation, HyDE plus cross-encoder reranking, and oracle context (the LLM receives the gold supporting passage directly). The six generator models (6×3 modes $= 18$ configurations) are local Llama-3.1-8B-Instruct, Llama-3.3-70B-Instruct, DeepSeek-V4-Pro, Claude-4.6-Sonnet (Anthropic), GPT-4.1 (OpenAI), and a production LLM-routing system (Atlas, our own previously-shipped product). All LLMs are accessed through a uniform OpenAI-compatible endpoint at zero infrastructure provisioning cost; Hyphae runs locally on commodity hardware (Apple Silicon laptop; Intel Xeon Platinum 8168 server).

1.1 Contributions

1. **Verifiable provenance as an addable layer, not a property of any one system.** The paper's object of study is the (verbatim-emission + SHA-256 hash-chained journal) layer: any system that emits byte-identical quotations of stored fragments over such a journal can have every output independently audited back to named, unaltered sources, without re-running the system. We show this property is *realizer-independent* — it belongs to the storage layer, is shared identically by Hyphae and by a trivial echo+journal baseline, and is absent from both plain echo and every LLM-RAG pipeline (Section 5.2). Hyphae is our worked instance of a system carrying the layer, not the source of the guarantee. The cryptographic mechanism (hash-chaining) is classical; our contribution is the observation that paraphrastic generation destroys the byte-bindability the audit relation requires, while extractive emission preserves it — which is what divides auditable from non-auditable retrieval systems. Equivalently, the layer makes citation *faithfulness* — whether the answer genuinely derives from the cited source rather than post-rationalising from a parametric prior, a distinction Wallat et al. [2024] show is both real and frequently violated in attributed LLM generation — trivially true by construction: a realizer with no parametric prior, whose answer *is* the cited fragment, cannot post-rationalise. It does not make answers more *correct* (the echo control forecloses that); correctness is realizer-independent, while faithfulness is exactly what verbatim emission structurally guarantees.

2. **An echo baseline that bounds what the standard benchmark can measure.** We include a trivial control — emit the retrieved seed sentence verbatim, no model, no lexicon, no composition — and evaluate it identically. On both corpora echo *ties or exceeds* Hyphae

on every comparable metric: gold-answer match, unsupported-claim rate, n-gram overlap, and verbatim-pass rate. This is the paper's pivotal negative result: measured correctness and grounding are properties of verbatim quotation *per se*, which any echo possesses, not of Hyphae's composition machinery. The standard correctness benchmark cannot distinguish Hyphae from a `print` statement, because the property that does — verifiable provenance — is off-axis for every metric in it.

3. **A minimal provenance benchmark that locates the property precisely.** We test the layer adversarially against the real journal across four tampering modes (edit, delete, insert, reorder) and two adversaries (store-only, chain-aware). The layer detects and localises 100% of store-only tampering — and so does echo+journal, identically — while plain echo and LLM-RAG detect 0%, and a chain-aware attacker who rewrites the persisted head defeats the bare chain. We close that boundary: an implemented external head anchor (Ed25519 signature held outside the store, ADR-0032) catches the chain-aware attack, restoring detection for any attacker who does not also hold the anchor key. The benchmark thus establishes the guarantee, its boundary, and the fix that discharges it, and confirms the dividing line is journal-vs-no-journal, not Hyphae-vs-the-rest.

4. **A two-axis correctness comparison against 18 LLM-based configurations, read as a sanity check rather than a win.** We score every system on an NLI-grounded unsupported-claim rate and a gold-answer match rate, across vanilla RAG, HyDE plus cross-encoder reranking, and oracle context. The result establishes that verbatim emission is *correctness-neutral*: it does not sacrifice gold-answer correctness relative to either echo or frontier LLMs in oracle mode (0.840–0.960). It does not establish that Hyphae is *more* correct, and we explicitly retract any such reading — the echo control forecloses it. The one architectural correctness effect that is real — verbatim emission removing the paraphrase-away failure mode LLM oracles still exhibit — is shared by echo, not unique to Hyphae.

5. **Latency and footprint as corollaries of not generating.** The realizer runs in 0.002–0.024 ms mean per query; the LLM baselines run in 1,858–6,083 ms whether hosted locally or via API (network round-trip $\sim 30\%$, Section 5.5), at ~ 50 MB versus multiple GB of resident memory. The five-to-six-order-of-magnitude gap is real and matters for deployment economics, but it follows from the decision not to run a generative model and is shared with the echo baseline — a corollary, not independent evidence for Hyphae.

6. **Component ablations and a hardware matrix consistent with the reframing.** We disable four Hyphae components in turn (cascade-shape composition, ethics gate, lexicon scale, boundary smoothing); only lexicon scale moves a correctness metric, and only marginally. We present this as corroboration, not independent proof — at $N = 34$ the ablations are underpowered, and the clean load-bearing evidence is the echo control (Section 5.1). Crucially, none of the four components, and neither the echo nor echo+journal baseline, perturbs the provenance layer: it is the one axis invariant across every configuration in the paper. The matrix re-runs on a server-class CPU-only instance, where quality metrics are hardware-invariant within ± 0.03 and the latency ratio widens from $\sim 1.9 \times 10^5$ to $\sim 9.3 \times 10^5$.

1.2 Roadmap

Section 2 positions Hyphae against the relevant literature on retrieval-augmented generation, structured memory systems, and NLI-based grounding metrics. Section 3 describes the architecture, with emphasis on the verbatim-quotation commitment that underlies the empirical results. Section 4

documents the experimental protocol — the two corpora, the six generator models, the three retrieval modes, the metric set, and the hardware. Section 5 presents the headline rankings on both corpora. Section 6 disaggregates HYPHAE's contribution by component. Section 7 discusses limitations (metric, corpus, statistical power, reader preference) and threats to validity. Section 8 concludes. The appendix contains full per-system result tables and reproduction instructions.

1.3 Reproducibility

The complete pipeline (HYPHAE source, comparator code, corpora, result envelopes, plotting code, and this paper's LaTeX source) is committed at `https://github.com/terrizoaguimor/hyphae-v2`. Each result JSON envelope carries the model identifier, hardware metadata, decoding hyperparameters, and per-query trace. The TriviaQA subset is regeneratable from the original dataset under random seed 42 via the project's `corpus_external.py` CLI. Total cost of reproducing the full experimental matrix at the LLM API layer is approximately USD 4 in inference tokens. The HYPHAE side has no cloud cost.

2 Related Work

2.1 Retrieval-Augmented Generation

The dominant pattern for grounding LLM responses in external knowledge couples a retrieval stage (chunking, embedding, top-k nearest-neighbour search) with a generation stage (LLM conditioned on the retrieved passages) [Lewis et al., 2020, Karpukhin et al., 2020]. Substantial subsequent work improves the retrieval side: HyDE generates a hypothetical answer with the LLM and embeds that, recovering performance when the raw query is short [Gao et al., 2022]; cross-encoder reranking re-scores candidates with $(query, passage)$ pairs scored directly [Nogueira and Cho, 2019, Reimers and Gurevych, 2019]; RAG-Fusion generates multiple query rewrites and fuses retrieval results with reciprocal rank fusion [Rackauckas, 2024]; GraphRAG constructs an offline graph of the corpus and traverses it at retrieval time [Edge et al., 2024]. Self-RAG trains the LLM to adaptively invoke retrieval [Asai et al., 2024].

These techniques operate on the retrieval pipeline; the generation stage remains an LLM. We compare HYPHAE against the vanilla pipeline, HyDE plus cross-encoder reranking (the canonical "serious RAG" stack), and oracle retrieval (the LLM receives the ground-truth supporting passage directly). The latter isolates composition quality from retrieval quality.

2.2 Attributed and Quote-Grounded Generation

A parallel line keeps the LLM in the answer path but makes the output *attributable*. Citation-generation systems emit an answer with citation markers and are evaluated by frameworks such as ALCE [Gao et al., 2023], whose citation-NLI metric scores whether the cited documents support the claim. Menick et al. [2022] (GopherCite) attaches *verbatim* supporting quotes to generated answers, learning the answer-to-quote support via reinforcement learning from human preferences; "according to" prompting [Weller et al., 2023] steers a model toward quoting its pre-training data and measures the tendency with an n-gram-overlap (QUIP) statistic. These establish verbatim quotation as an attribution device as known prior art: Schuster et al. [2023] (SEMQA) make it explicit, defining a *semi-extractive* task whose answers interleave verbatim source spans with generated connectors for "by-design" easy-to-verify attribution.

HYPHAE differs from all of these on one structural axis: *there is no model output to attribute.* The answer is not generated and then linked to evidence; the answer *is* the stored fragment, emitted byte-

for-byte, with the binding made cryptographically tamper-evident rather than learned (GopherCite), prompted (according-to), or textual-only (SEMQA, ALCE). This collapses the distinction Wallat et al. [2024] draw between citation *correctness* (the cited source supports the claim) and citation *faithfulness* (the model genuinely relied on the source rather than post-rationalising from its parametric prior, a failure they find afflicts a large fraction of attributed answers): a system with no parametric prior, whose answer *is* its source, is faithful by construction — the post-rationalisation failure mode is structurally eliminated, not merely reduced. It does not collapse *correctness*, which remains a retrieval-relevance question; Worledge et al. [2024] characterise the resulting extractive–abstractive verifiability trade-off, with the byte-identical pole maximally verifiable and least abstractive. This is consistent with our echo control: correctness is realizer-independent, faithfulness is what verbatim emission makes trivially true. Our provenance benchmark (Section 5.2) is correspondingly orthogonal to attribution benchmarks like ALCE — it scores cryptographic tamper *detection and localisation* over the journal, not citation correctness over the answer.

2.3 Structured Memory and Cognitive Architectures

Long-context architectures and explicit memory systems offer different approaches to persistent context. MemGPT manages context windows via a tiered storage system with LLM-mediated paging [Packer et al., 2024]; Atlas (the production system behind our `router:celiums-conversation` comparator) routes queries across a curated model ensemble. Earlier work on cognitive architectures (Soar [Laird, 2012], ACT-R [Anderson, 2007]) modelled memory and inference as discrete operations on structured representations rather than as prediction over token distributions.

Hyphae shares the structured-memory orientation but differs in its composition stage: the realizer is not an LLM, not a learned model, and not a statistical predictor. It is a finite procedure walking a curated lexicon and emitting verbatim quotations of retrieved fragments. The architecture is finite and inspectable; no training run can change its outputs.

2.4 Verifiable AI and NLI-Based Grounding

Recent work on factual grounding evaluates LLM outputs against their source contexts using natural language inference (NLI) models [Honovich et al., 2022, Gekhman et al., 2023]. The `unsup_f` metric we employ is in this family: each sentence in a response is scored for entailment against the retrieved context, and sentences classified `neutral` or `contradiction` count as unsupported. Recent surveys [Huang et al., 2023, Ji et al., 2023] catalogue the failure modes such metrics catch and miss.

We employ the `roberta-large-mnli` cross-encoder [Liu et al., 2019] for entailment scoring. Section 4 documents the metric's known limitations (asymmetric treatment of hedging, sensitivity to scaffolding prose) and how Hyphae's compositional template interacts with them.

2.5 Sub-millisecond Production Inference

The latency-quality trade-off in conversational AI is widely acknowledged, and a large line of work attacks LLM inference cost directly — e.g. speculative and staged-speculative decoding [Leviathan et al., 2023, Chen et al., 2023]. Most such work targets sub-second response under high token generation budgets. Hyphae sits in a different operating point: sub-millisecond composition under the structural constraint that the response is bounded by curated lexicon phrases and verbatim quotations. This shifts the operating point of the system at the cost of generative flexibility that some applications require and others do not.

2.6 Tamper-Evident Logs and Provenance

The cryptographic mechanism underlying HYPHAE's journal — a hash chain linking each record to its predecessor by digest — is classical, and we are explicit about that. Linked timestamping by hash-chaining dates to Haber and Stornetta [1991]; Merkle trees [Merkle, 1988] give the same tamper-evidence with logarithmic verification; Certificate Transparency [Laurie et al., 2013] deploys append-only Merkle logs at internet scale with external auditing; and content-addressed DAGs such as Git [Torvalds et al., 2005] make the construction everyday infrastructure. We claim none of this machinery as novel. What HYPHAE's journal adds over a bare hash chain is only what every audited store needs (durable persistence, an ordered chain over fragment ingests); the contribution of this paper is not the chain.

Our contribution is an *observation* about where such a chain can and cannot be attached in a generation system: the audit relation requires that the emitted output be byte-bindable to a journalled source, and **paraphrastic generation destroys that bindability** while **extractive (verbatim) emission preserves it**. An LLM-RAG pipeline can hash-chain its corpus all it likes; because its output paraphrases the retrieved text, no output span is byte-identical to any journalled fragment, and the chain cannot bind the answer the user received to the source it came from. This is the property that divides auditable from non-auditable retrieval systems, it is orthogonal to retrieval quality and to the choice of hash construction, and to our knowledge it has not been stated as the operative criterion for provenance in retrieval-augmented generation.

2.7 Where This Paper Fits

We do not claim that HYPHAE subsumes or replaces LLM-based generation, and we do not claim it is more correct: our echo control (Section 5.1) shows that on standard correctness and grounding metrics a verbatim `print` of the retrieved sentence matches it, so those metrics measure verbatim quotation rather than the architecture. We claim instead that *for the subset of tasks in which the unit of grounding is a discrete retrieved fragment and the deployment requires provable provenance*, HYPHAE provides a property no LLM-RAG pipeline and no echo baseline provides by construction: every emitted span is byte-identical to a fragment in a hash-chained journal, so the output is tamper-evident under the threat model of Section 5.2 — auditable back to named, unaltered source fragments against any attacker who does not hold the external head-anchor key, which we implement and demonstrate. The trade-off, made explicit, is that the prose is template-rigid; users wanting LLM-style paraphrastic composition need an LLM.

The contribution is therefore positioned as a verifiable *provenance layer* for grounded retrieval, not a correctness-competitive replacement for LLM-RAG.

3 The Hyphae Architecture

HYPHAE is structured as three concentric layers: **substrate** (memory state, fragment storage, hash-chained journal), **cognition path** (the operations that retrieve, compose, and emit), and **realizer** (the prose-emission stage). The single property that distinguishes HYPHAE from every baseline in this paper — including the trivial echo baseline that matches it on all correctness metrics — is the *audit relation* between the realizer's output and the substrate's hash-chained journal: every emitted quotation is byte-identical to a journalled fragment, and that correspondence is independently checkable. We describe each layer with that property in focus, and return to it as the core of the contribution in Section 3.4.

3.1 Substrate

The substrate stores *cognitive fragments*: small self-contained units of content with provenance metadata (source subsystem, parent fragments if cascade-derived, confabulation risk score, valence, domain tags). Fragments are written to disk via an embedded key-value store [Hesterberg, 2024, Jablonski, 2024] and chained into a SHA-256 hash sequence that links each fragment to its predecessor by content hash. The journal is verifiable: any modification to a previously- written fragment breaks the chain at and after the modification point.

Retrieval is via approximate-nearest-neighbour search [Malkov and Yashunin, 2018] over fragment embeddings produced by a deterministic hashing token embedder. The embedder is not an LLM, contains no learned parameters, and uses the `HashingTokenEmbedder` primitive from the substrate. This makes the retrieval stage deterministic and reproducible across runs.

3.2 Cognition Path

A **cognition path** is a procedure that reads from the substrate, performs a structured operation, and either writes new fragments back or emits a response through the realizer. The v0.1 substrate implements five paths: `ingest` (write a new fragment), `recall` (retrieve fragments similar to a cue), `compose` (synthesize a response from retrieved fragments), `learning_update` (adjust substrate-internal parameters under audit), and `grounded_retrieval` (deferred per the v0.1 RFC).

The `compose` path is the one this paper measures. It takes a *working set* of cognitive fragments (typically retrieved by a `recall` call) and produces a `RealizationOutput`: composed prose, a list of fragment IDs quoted in emission order, and a structured set of limitation triggers (such as `HighConfabRisk` or `ShallowCascade`) that fired during composition.

3.3 Realizer

The realizer is the surface-prose-emission stage. It is the component where the LLM would conventionally sit. In Hyphae it is a finite procedure:

1. **Schema selection.** The caller's intent (one of: dialogue, assert, summarize, compare, reflect, narrate) maps deterministically to a schema (DialogueReply, GroundedAssertion, Summary, ComparativeAnalysis, IntrospectiveAssessment, NarrativeArc).

2. **Shape derivation.** If the caller supplies an explicit composition shape, the realizer walks its steps. Otherwise the realizer derives a shape from the working set by detecting opposed valence (Contrast role) and adjacency relations (Continuation role). The shape determines the sequence of connective roles between adjacent quotations.

3. **Connective selection.** For each inter-fragment boundary, the realizer queries the lexicon for a connective matching the (role, register, polarity, formality) tuple. Boundary smoothing rules (Section 3.5) filter out connectives whose head would clash with the adjacent quote's tail.

4. **Quote emission.** Each fragment's body is emitted verbatim inside double quotes. No paraphrase, no summarisation, no reordering of intra-body text.

5. **Limitation acknowledgement.** Triggers that fired during composition (empty working set, high confabulation risk, shallow cascade depth, ethically sensitive content) are surfaced via the lexicon's limitation-acknowledgement phrases.

The realizer's output is bounded: every emitted token comes either from the curated lexicon (approximately 250 phrases covering 10 connective roles, 4 conversational registers, 4 polarity classes, and 3 formality tiers) or from a fragment body quoted verbatim. No statistical model decides the emission.

3.4 Hard Commitment 12 — Verbatim Quotation

The architectural decision underlying the empirical results is *Hard Architectural Commitment 12* from the project's foundational ADR (`docs/adr/0001-fresh-from-v1.md`):

> *Composition uses fragment quotation plus connective tissue, not novel language synthesis. Fragments are opaque content sources whose body text is preserved verbatim. The surface realizer generates only the structural prose connecting them. This boundary is load-bearing for the no-LLM-in-cognition-path claim.*

This commitment trades expressive flexibility for two properties that anchor the empirical comparison in Section 5:

Verifiability. Every quoted body in a HYPHAE response is byte-identical to a fragment stored in the substrate, which is itself verifiable against the hash-chained journal. The audit relation is total: a third party can independently confirm that the output was constructed from named substrate fragments without re-running the system.

Unsupported-claim immunity by construction. The NLI metric we employ (Section 4.4) cannot mark a verbatim quote as unsupported with respect to its source: the quote and the context are identical text. The connective tissue between quotes is non-factual scaffolding the metric's filter recognises and excludes from the denominator. The result is that on factual-retrieval tasks, the architectural commitment maps directly onto the metric's scoring shape.

3.5 Lexicon and Boundary Smoothing

The lexicon is a curated mapping from *(role, register, polarity, formality)* tuples to surface phrases. The baseline EN lexicon contains approximately 250 entries organised across 10 connective roles (Opening, Continuation, Contrast, Attribution, Closing, Concession, Causation, Elaboration, Sequence, Summary), 4 register classes (Neutral, Formal, Conversational, Technical), 4 polarity classes (Continuation, ContrastSoft, ContrastHard, Concession, Neutral), and 3 formality tiers (Low, Mid, High).

The picker has a four-level fallback chain: an exact $(role, register, polarity, formality)$ tuple match is preferred; on miss, formality relaxes, then register, then polarity, until finally any phrase for the requested role is returned.

Boundary smoothing avoids surface-level disfluencies at the boundary between a connective and an adjacent fragment quote. For example, the connective *"Per the recorded fragments,"* followed by a quote that begins with *"Per the"* would produce a stuttered repetition. The smoothing pass extracts boundary signals from the adjacent quote (leading determiner, anaphor surface form) and filters the candidate connective set to avoid such collisions.

Section 6 reports ablations of cascade-shape composition, ethics gate, lexicon scale, and boundary smoothing. Of these, only lexicon scale produces a measurable effect on the comparator metrics — a finding that constrains claims about the contribution of the other three components at our corpus size.

3.6 What the Realizer Does Not Do

The realizer does not paraphrase, summarise, reorder intra-body text, infer over multiple fragments, or generate prose outside the lexicon's curated set. These restrictions are not implementation gaps; they are architectural decisions. The choice trades prose-fluency dimensions in which an LLM excels for the dimensions catalogued above.

For applications requiring paraphrastic flexibility or synthesis over multiple sources, Hyphae's realizer is not the right tool. The right comparator question is therefore not "does Hyphae replace an LLM," but rather "for the subset of tasks where verbatim grounding suffices, is the substrate competitive?". Sections 5 and 7 take up that question.

4 Experimental Setup

4.1 Two Corpora

We evaluate every system on two corpora.

Own corpus (N=34). Curated for the project, organised by intent and schema (4 dialogue queries with cascade-derived seeds, 2 grounded-assertion queries, 2 empty-working-set queries, 2 high-confabulation-risk queries, 1 shallow-cascade query, 1 valence-opposed query, plus three fluency-exercise queries from ADR-0008, ten bucket-coverage queries from ADR-0009 across Conversational/Formal/Neutral/Mixed registers, and nine schema-coverage queries from ADRs-0016, 0023, 0024, 0025 across Summary, ComparativeAnalysis, IntrospectiveAssessment, and NarrativeArc). The corpus is concentrated on engineering and deployment scenarios reflecting the project's primary intended deployment context. Source: `crates/hyphae-eval/src/corpus.rs`.

TriviaQA subset (N=150). Random sample (seed 42) from the TriviaQA `rc` validation split [Joshi et al., 2017]. For each sampled query, we extract a seed body from the supplied Wikipedia context by (1) filtering to sentences containing the answer or any alias as a word-bounded match, (2) restricting to sentences of 30–250 characters, and (3) reranking the survivors by cosine similarity to the query using the `sentence-transformers/all-MiniLM-L6-v2` embedder. The highest-scoring sentence becomes the seed body. The filter rejection rate on this seed is approximately 17% (21 samples without wiki_context, 10 without a qualifying sentence) yielding 150 surviving queries. Source: `bench/baseline-llm-rag/src/baseline_llm_rag/corpus_external.py`.

The two corpora are complementary. The own corpus tests multi-fragment composition (working sets of 2–3 seeds with varied registers); TriviaQA tests single-fragment retrieval. The own corpus is dense with engineering scenarios; TriviaQA spans biography, geography, history, science, pop culture, and sports.

4.2 Generator Models

We compare Hyphae against six LLM-based configurations spanning the model-class landscape:

- **Llama-3.1-8B-Instruct Q4_K_M** (4.6 GB GGUF): small open model, run locally via `llama-cpp-python` on the laptop's Metal backend or a DigitalOcean CPU droplet. Establishes the small-open baseline.
- **Llama-3.3-70B-Instruct** (FP8 hosted): larger open model, accessed via DigitalOcean Inference [DigitalOcean, 2025]. Tests whether scale helps.
- **Claude-4.6-Sonnet** (frontier closed, Anthropic): tests a frontier model from a different lab family.

- **GPT-4.1** (frontier closed, OpenAI): tests a frontier model from the family most-frequently cited as the LLM baseline.
- **DeepSeek-V4-Pro** (frontier open reasoning): tests a frontier open model with reasoning training.
- **router:celiums-conversation** (our in-house Atlas router): tests a production LLM-routing system configured for conversational dispatch.

All six are accessed through a uniform OpenAI-compatible API endpoint provided by DigitalOcean's GenAI Platform, with the exception of the local Llama-8B which uses `llama.cpp`. This uniformity removes API-shape variance from the comparison while preserving the providers' native decoding pipelines.

4.3 Retrieval Modes

Each LLM is evaluated under three retrieval modes:

- **Oracle.** The corpus's seed bodies are passed directly as context. No retrieval; the LLM receives the gold supporting passage. Isolates composition quality from retrieval quality.
- **Vanilla RAG.** FAISS [Johnson et al., 2019] `IndexFlatIP` (exact cosine after L2 normalisation) over chunks of all seed bodies. Top-$k = 5$ retrieved, passed as context. Chunking uses `tiktoken cl100k_base` BPE with 256-token chunks and 32-token overlap.
- **Strong RAG.** HyDE plus cross-encoder reranking. The LLM first generates a hypothetical answer; that answer is embedded and retrieved (over-retrieve $k = 20$); the candidates are reranked by the `BAAI/bge-reranker-base` cross-encoder [BAAI, 2024] and the top-5 passed as context.

The retrieval pipeline (embedder, chunker, FAISS index, reranker) is shared across all six LLMs. Only the generation stage varies.

4.4 Metrics

We score every system on the same set of metrics, comprising those directly comparable across architectures (the *comparable subset*) and two extra metrics added to capture properties specific to the verbatim-quotation commitment.

Comparable subset.

- `verbatim_pass` (boolean): every seed body marked `verbatim_quotation` appears verbatim in the response.
- `connective_hygiene_pass` (boolean): the response is free of doubled-connective stutters from a pinned list.
- `ngram_overlap_`n ($n \in \{4, 5, 8\}$): the fraction of n-grams in the response that appear in the concatenated retrieved chunks after lowercasing and whitespace normalisation.

- `unsup_f` (NLI-based): each response is split into sentences. Sentences beginning with a known connective phrase from a pinned list are excluded as scaffolding. The remaining factual sentences are scored by `roberta-large-mnli` as entailment / neutral / contradiction against the concatenated retrieved context. The filtered rate is (neutral + contradiction) / (total factual).
- `unsup_r`: the same as `unsup_f` without the connective- sentence filter. Higher than `unsup_f` for any system whose output contains scaffolding phrases.
- `latency_p50`, `latency_p95`, `latency_mean`: per-query wall-clock latency aggregated over the corpus.

Verbatim-quotation extras.

- `quoted_content_supported_rate`: the fraction of double-quoted spans in the response that appear verbatim in the retrieved context. Hyphae produces quoted spans for every fragment; LLM-based systems rarely use double quotes formally.

Hyphae-only diagnostics. HYPHAE produces a structured `RealizationOutput` carrying limitation triggers, schema identifier, and fragments-quoted list. The internal scoring harness measures nine additional dimensions over this output (schema match, limitation recall/precision, lexical diversity, role coverage, boundary smoothness). These are inapplicable to LLM-based systems and are reported separately, not in the head-to-head table.

Statistical inference. For every aggregate metric we report a bootstrap percentile confidence interval (1000 resamples, 95% level, seed=42). At $N = 34$ (own corpus) the CIs are wide; at $N = 150$ (TriviaQA) they narrow substantially. The headline results we emphasise survive this narrowing.

4.5 Hardware

The primary experimental hardware is an Apple Silicon laptop (M-series, 10 cores, MPS backend for the local Llama-8B and the NLI scorer). Network calls to DigitalOcean Inference run on the laptop's network.

For the hardware-matrix portion (Section 5.6) we provisioned a DigitalOcean `c-16` CPU-optimised droplet (Intel Xeon Platinum 8168, 16 dedicated vCPU, 31 GB RAM, Ubuntu 24.04 LTS, NYC1 region) and re-ran the local Llama-8B matrix and the HYPHAE ablations there.

Latency reporting convention. Two conventions are used and must be read together. **Tables 1 and 3 report** $p50$ (median per-query latency); **Tables 4 and 5 report means**. The two differ because each corpus's per-query latency distribution is right-skewed (a few long LLM generations pull the mean above the median), so the same configuration legitimately shows, e.g., Llama-8B oracle at 1,598 ms ($p50$, TriviaQA, Table 1), 1,858 ms (mean, TriviaQA, Table 4), and 2,299 ms (mean, own corpus, Table 5). These are not contradictory; they are median-vs-mean and TriviaQA-vs-own-corpus. We use means in the latency-focused tables because the headline ratio is a mean-over-corpus quantity. HYPHAE's per-query latency sits at the microsecond timer-resolution floor; its $p50$ rounds below 0.1 ms and is not individually meaningful, so we report HYPHAE latency only as a corpus mean (0.002 ms on TriviaQA's single-seed queries, 0.024 ms on the own corpus's multi-fragment queries, 0.007 ms on the Xeon droplet) and treat all HYPHAE latency figures as order-of-magnitude (tens of microseconds or below), not precise measurements.

The reported HYPHAE latencies should be interpreted as a lower bound on what is achievable: optimisation work has not been prioritised. The reported LLM latencies should be interpreted as the best plausible API-routed latency under no batching — production systems with batching and rate-limited parallelism may achieve different throughput, but the per-query latency on the user-perceived path is bounded below by the LLM's serial inference time.

5 Results

5.1 TriviaQA Standard Benchmark

Table 1 reports the system ranking on the TriviaQA 150-query subset on **two complementary axes**: **gold-answer match** (`gold`, does the response contain the benchmark's reference answer or any alias, with word-bounded matching?) and **NLI-grounded unsupported-claim rate** (`unsup_f`, does the response stay inside the retrieved context?). The first axis measures correctness; the second measures grounding.

Read the echo control first. The top row of Table 1 is the **echo baseline**: emit the retrieved seed sentence verbatim, with no model, no lexicon, and no composition. It reaches 1.000 gold-answer, 0.000 `unsup_f`, 0.000 `unsup_r`, and 1.000 n-gram overlap — it ties HYPHAE on gold-answer and `unsup_f`, and *exceeds* it on `unsup_r` and overlap, because HYPHAE's connective scaffolding ("Drawing from working memory, . . . ") is the only material in HYPHAE's output that the NLI can score as non-entailed or that dilutes the n-gram match. The consequence is stark and is the organising fact of this paper: **a one-line `print` of the retrieved sentence is at least as good as Hyphae on every metric in this table.** Whatever HYPHAE contributes over echo — multi-fragment composition, register-appropriate connectives, and most importantly the hash-chained provenance relation — is not visible to, or is penalised by, this metric suite. The remainder of this section should therefore be read not as "HYPHAE wins" but as "HYPHAE does not *lose* correctness relative to a trivial echo or to frontier LLMs, while adding an audit property neither has."

Why Hyphae's gold-answer is 1.000. The TriviaQA corpus column was constructed by the converter at `corpus_external.py` by sampling validation queries whose Wikipedia context contained the gold answer, and selecting as the seed body the sentence most semantically similar to the query that contained the answer (or any alias). By construction, every seed body the realizer is given contains the gold answer. HYPHAE quotes the seed body verbatim, so the gold answer always appears in HYPHAE's response. The 1.000 rate is therefore a property of the corpus construction interacting with the verbatim-quotation commitment, not an unconditional empirical claim. We flag this explicitly, and one further limit deserves emphasis: this parity holds only because the seed is guaranteed to contain the answer. On a corpus where retrieval can fail to surface an answer-bearing fragment, HYPHAE would be *retrieval-capped* exactly like echo — it can only quote what retrieval delivers, and its deterministic hashing embedder is a weaker retriever than the dense retrievers the LLM baselines use. The gold-answer comparison should thus be read as "given a correct retrieval, verbatim quotation does not lose the answer the way generation sometimes does" (the LLM oracle drop to 0.84–0.96), not as a claim about end-to-end QA accuracy, where retrieval quality would dominate and HYPHAE has no advantage.

LLM gold-answer rates on this corpus. The frontier LLMs given the same seed bodies in *oracle* mode (no retrieval, the gold-context is the input) reach 0.840 (DeepSeek-V4-Pro) to 0.960 (Claude-4.6-Sonnet) gold-answer match. That gap from 1.000 is informative: **even when the**

Table 1: **TriviaQA-150 ranking, with the echo control on top.** `gold` = gold-answer match rate (correctness). `unsup_f` = NLI-grounded unsupported-claim rate (grounding). `unsup_r` = same NLI metric without the connective-sentence filter. The *echo baseline* (verbatim emit of the retrieved sentence, no model) ties HYPHAE on gold-answer and `unsup_f` and exceeds it on `unsup_r` and overlap — the benchmark cannot separate HYPHAE from a `print` statement. HYPHAE's 1.000 gold-answer holds by construction of the corpus filter (see text). The LLM configurations reach 0.84–0.96 gold-answer at 0.400–0.737 `unsup_f`: nearly as *correct* as both echo and HYPHAE, scored less *grounded* by the NLI metric for reasons of phrasing, not of whether they answer correctly. `router:celiums` is the authors' own production LLM router (Atlas), included for self-comparison and labelled as such.

Rank	System	gold	unsup_f	unsup_r	overlap$_4$	lat$_{p50}$ (ms)
–	*Echo baseline (control)*	*1.000*	*0.000*	*0.000*	*1.000*	<0.1
1	**Hyphae**	**1.000**	**0.000**	0.013	0.600	<0.1
2	Claude-4.6-Sonnet oracle	0.960	0.623	0.651	0.123	2,363
3	Claude-4.6-Sonnet rag	0.933	0.606	0.635	0.155	2,509
3	router:celiums rag	0.933	0.737	0.756	0.161	1,119
5	Claude-4.6-Sonnet strong-rag	0.913	0.618	0.649	0.147	5,349
6	Llama-3.3-70B oracle	0.907	0.664	0.664	0.121	1,905
6	router:celiums strong-rag	0.907	0.729	0.741	0.153	2,310
8	Llama-3.3-70B rag	0.900	0.711	0.711	0.137	2,253
8	GPT-4.1 oracle	0.900	0.644	0.654	0.100	916
10	router:celiums oracle	0.887	0.585	0.601	0.187	972
11	Llama-3.3-70B strong-rag	0.867	0.701	0.712	0.135	6,107
12	GPT-4.1 rag	0.853	0.597	0.607	0.167	925
13	DeepSeek-V4-Pro oracle	0.840	0.400	0.406	0.504	647
14	GPT-4.1 strong-rag	0.813	0.620	0.635	0.137	2,165
15	DeepSeek-V4-Pro rag	0.793	0.613	0.621	0.332	651
16	Llama-8B-local oracle	0.700	0.664	0.648	0.168	1,598
17	DeepSeek-V4-Pro strong-rag	0.693	0.630	0.641	0.300	1,787
18	Llama-8B-local rag	0.500	0.783	0.777	0.178	3,220
19	Llama-8B-local strong-rag	0.500	0.783	0.771	0.182	9,176

answer is in the supplied context, an LLM occasionally paraphrases it away, summarises in a way that elides the entity, or hedges into a non-committal response. Llama-8B in oracle mode reaches only 0.700. The verbatim-quotation commitment removes this failure mode by construction.

Why Hyphae's `unsup_f` is 0.000**.** HYPHAE's response on a single-seed query is illustrated in Listing 1: one verbatim quote (trivially entailed against the retrieved context because the quote *is* a substring of the context) plus two scaffolding sentences. Both scaffolding sentences begin with a recognised connective prefix and are excluded from the unsupported-claim denominator by the `is_connective_sentence` filter. The NLI denominator is zero; the rate is zero by quotient. We report `unsup_r` (the unfiltered version) alongside `unsup_f` for honesty: HYPHAE's `unsup_r` on TriviaQA is 0.013 rather than 0.000, reflecting the small fraction of queries where the splitter and the connective filter disagree at the boundary between scaffolding and quote.

```
Drawing from working memory, "Pearson did not make the move into
politics until a few years later, after King had announced his
retirement as the Prime Minister of Canada." That is what working
```

```
memory holds on this.
```

Listing 1: HYPHAE response on TriviaQA query `triviaqa-qb_4397`.

Why LLM `unsup_f` is high even at high gold-answer. LLM responses to TriviaQA queries typically include meta-claims about the source ("According to the context," "As stated in the sentence," "This information directly answers the question about…"). The NLI scorer marks these `neutral` (they describe the relationship between text and question, not facts about the world). The connective filter does not catch them because they do not begin with a known connective prefix. They count as unsupported in the filtered metric. A Claude response that contains the gold answer alongside several such meta-claims reaches 0.960 gold-answer but 0.623 `unsup_f`. The metrics disagree because they measure different things; both numbers are correct.

The honest comparison. The one correctness mechanism that is real and architectural — visible in the gold-answer column — is that verbatim quotation removes a paraphrase-away failure mode: in oracle mode, where every system receives the same gold-bearing seed body, the LLM oracle baselines still drop four (Claude) to thirty (Llama-8B-local) percentage points of gold-answer correctness by paraphrasing the answer out of their response, while HYPHAE and the echo baseline cannot. But note the company HYPHAE keeps in that finding: *the echo baseline shares it exactly.* Verbatim emission is what removes paraphrase-away, and echo is verbatim emission with nothing else. The NLI grounding gap, meanwhile, is on this corpus predominantly a metric artefact (documented via the filtered/unfiltered rates and the samples in Appendix C). Neither the correctness axis nor the grounding axis isolates anything HYPHAE does that echo does not. What does — the hash-chained provenance relation (Section 3.4) — is not measured by either axis, and is the subject of Section 7.

5.2 Provenance: a Minimal Tamper-Detection Benchmark

The echo control shows the correctness benchmark cannot distinguish HYPHAE from a `print`. This subsection measures the property that can — and, just as importantly, locates it precisely. The property is *verifiable provenance*: a hash-chained journal over the stored fragments makes post-ingest store tampering detectable and localised. We test it adversarially against the real `hyphae_storage` journal (not a mock), crossing two axes: **tampering mode** and **adversary capability**.

The property is realizer-independent — it is not Hyphae's. This is the central honesty of the section. The journal stores *raw fragment bodies*; the experiment invokes no lexicon, no cascade-shape composition, no realizer at all. Whatever the detection result is, it is a property of the (verbatim-store + hash-chain) layer, and it is therefore shared identically by HYPHAE and by an **echo+journal** baseline — echo with the same journal bolted on. Exactly as the echo control showed correctness is not HYPHAE-specific, this shows provenance is not HYPHAE-specific either: it is an *addable layer*, and HYPHAE is one worked instance of a system that carries it. What both lack is captured by what echo (*without* a journal) and LLM-RAG lack.

Protocol. We ingest N fragment bodies via the production `append` path and confirm `verify()` passes. We then apply one tampering mode and re-verify, recording detection and the localised sequence. Two adversaries: **store-only** (write access, edits records in place, does not reimplement the chain logic) and **chain-aware** (knows the hash construction; recomputes every hash forward from the edit and rewrites the persisted head).

Table 2: **Minimal provenance benchmark: post-ingest store-tampering detection.** The (verbatim + hash-chain journal) layer — shared identically by HYPHAE and a trivial echo+journal baseline, since the journal stores raw bodies — detects and localises every store-only tampering mode. A chain-aware adversary who rewrites the persisted head defeats the *bare* chain, but is caught once the head is externally anchored with an Ed25519 signature (ADR-0032, key held outside the store). Echo *without* a journal and LLM-RAG have no integrity layer at all; LLM-RAG is doubly exposed, as its output paraphrases the source (`verbatim_pass` 0.09–0.24, Tables 1–3) and so cannot even be string-matched back to a source for post-hoc audit.

System	store-only adv.	chain-aware (bare)	chain-aware (anchored)
Verbatim + journal (HYPHAE *and* echo+journal)	**100%** detect & localise all 4 modes	defeated (rewrites head)	**DETECTED** (Ed25519 head)
Echo (no journal)	0%	0%	—
LLM-RAG (no journal)	0%	0%	—

Result (Table 2). Against the store-only adversary the layer detects and localises **100%** of tampering across all four modes — edit, delete, insert, reorder — because any of them leaves some entry's stored `prev_hash` inconsistent with the recomputed chain (insertion of a forged trailing record is caught at that record; deletion and reorder break the successor link; the final-entry case is caught by the persisted head). The chain-aware adversary **defeats the bare chain**: having recomputed the chain and rewritten the head, the forged journal verifies cleanly. We close that gap by anchoring the head: an Ed25519 signature over the head, produced by a key the store process does not hold (ADR-0032), is published outside the store. The same chain-aware attack now **fails anchored verification** — the attacker cannot re-sign the rewritten head without the anchor key — so detection returns to 100% for any attacker who does not additionally compromise that key. We implement and run this; it is not a promissory note.

Threat model — what this does and does not show. The experiment demonstrates *tamper-evidence*, not *tamper-proofing*, and the distinction is load-bearing. We assume an attacker with **write access to the fragment store** who edits entries but **does not reimplement the chain logic** — the realistic store-level adversary who rewrites records in place. Against that attacker the hash chain is decisive: any edited entry's recomputed content hash no longer matches its successor's stored `prev_hash`, so `verify()` fails and localises the break. What the bare chain does *not* defend against is a stronger attacker who *does* know the chain logic: such an attacker can recompute every hash forward from the edit point and re-persist a chain that verifies cleanly. The entire security of the scheme therefore reduces to the integrity of one value — the persisted chain head. In this experiment the head is a local record, so the guarantee is "tamper-evident against an attacker who cannot forge the head". **Tamper-proofing in the full sense requires anchoring the chain head outside the attacker's write scope**: a signed Merkle root published to an external append-only ledger, a third-party timestamp, or simply a head stored on write-isolated media. The local hash chain is the correct base layer for that construction — it makes every entry's integrity checkable against the head — and external anchoring is a natural add-on, not a redesign. We state this explicitly because the high-level language of "cryptographic audit" can suggest a stronger guarantee than a *bare* local chain delivers on its own. The next paragraph closes exactly this gap with an implemented external anchor; the bare-chain analysis here is the baseline against which that closure is measured.

Where the head lives, and how anchoring closes the gap. The store-only and chain-aware adversaries are separated by exactly one assumption: that the persisted chain head is outside the attacker's write scope. In the bare configuration this assumption does *not* hold — the head is a record in the same `fjall` keyspace (the `meta` partition) as the fragment bodies, so an attacker with store write access can rewrite the head and the store-only adversary collapses into the chain-aware one. This was, in earlier drafts, the single load-bearing dependency of the positive result, and we treated it honestly as such. We now **discharge it rather than declare it**: external head anchoring (ADR-0032) signs the head with an Ed25519 key the store process does not hold and publishes the signature outside the store. A chain-aware attacker who rewrites the head to a recomputed value cannot produce a matching signature without the anchor key; anchored verification fails, and the attack is detected (Table 2, rightmost column). The guarantee therefore strengthens from "tamper-evident against an attacker who cannot write the head" (vacuous when the head shares the store) to "tamper-evident against an attacker who does not hold the anchor signing key" — realistic, because the key lives in an HSM / offline signer / audit service, whereas the head necessarily lives in the store. The residual assumption (the attacker does not compromise the signing key) is the standard key- management reduction every signed-log scheme carries; publishing anchors to an external append-only ledger for cross-snapshot freshness is the natural next step (Section 7).

Reading. This is the paper's positive result, and the echo control sharpens rather than supports Hyphae specifically. On correctness, echo ≡ Hyphae; on provenance, echo+journal ≡ Hyphae (the journal stores raw bodies, so the realizer is irrelevant to detection). The line that actually separates systems is not Hyphae-vs-the-rest but **journal-vs-no-journal**: a verbatim retriever that carries a hash-chained store can be audited; one that does not (plain echo) or that paraphrases its sources (LLM-RAG) cannot. The contribution is therefore the *addable provenance layer* — verbatim emission plus a tamper-evident journal — and Hyphae is a worked instance of it, not its sole embodiment. That framing is deliberately more modest and more defensible than "Hyphae is uniquely auditable". Scaling this demonstration into a community provenance benchmark, and anchoring the chain head externally to extend tamper-evidence to tamper-proofing, are the natural next steps (Section 7); the experiment here establishes that the base-layer property is real, measurable, and realizer-independent — not merely asserted.

5.3 Own Corpus

Table 3 reports the same 19-system ranking on the own 34-query corpus. The picture is different: GPT-4.1 with strong retrieval reaches rank 1 (`unsup_f`= 0.211), with Hyphae at rank 2 (`unsup_f`= 0.219), within the bootstrap 95% CI overlap.

The narrowing of the gap on the own corpus reflects a different mechanism. The own corpus's multi-fragment queries (e.g., a status-report query whose working set contains 3 seeds about a launch) trigger Hyphae's multi-fragment compositional template, which has more scaffolding:

```
Per the recorded fragments, "the migration completed at 14:02 UTC"
Per the next fragment, "the monitoring dashboards stayed green
for the hour after the cutover" That is the substrate's current
view.
```

Listing 2: Hyphae response on the own-corpus `dialogue-001` query.

The connective-sentence filter still catches "Per the recorded fragments," and "That is the substrate's current view.", but the inter-fragment connectives ("Per the next fragment,") sit mid-response without leading the sentence; the splitter recognises them as inter-fragment dividers but

Table 3: **Own corpus (N=34) ranking, with the echo control on top.** Sorted by `unsup_f`. The echo baseline again ties or beats every system, including Hyphae, on the comparable metrics — on this multi-fragment corpus the margin is wider (0.000 vs Hyphae's 0.219 `unsup_f`) because Hyphae's multi-fragment compositions carry more connective scaffolding for the NLI to score neutral. GPT-4.1 with strong retrieval is the strongest LLM here, within bootstrap CI of Hyphae on `unsup_f`. None of this is a correctness win for Hyphae; it is evidence that the metric rewards bare quotation, which echo maximises.

Rank	System	`verbatim_pass`	`unsup_f`	`unsup_r`	overlap_4	lat_{p50} (ms)
–	*Echo baseline (control)*	*1.000*	*0.000*	*0.000*	*1.000*	<0.1
1	GPT-4.1 strong-rag	0.176	0.211	0.210	0.468	2,314
2	**Hyphae**	**1.000**	0.219	0.625	0.466	<0.1
3	GPT-4.1 oracle	0.147	0.271	0.265	0.355	1,060
4	DeepSeek-V4-Pro strong-rag	0.206	0.300	0.300	0.405	2,759
5	GPT-4.1 rag	0.206	0.329	0.346	0.446	1,017
6	Llama-8B-local strong-rag	0.176	0.340	0.348	0.509	11,938
7	Llama-3.3-70B oracle	0.118	0.346	0.383	0.204	2,011
8	Llama-8B-local oracle	0.176	0.367	0.376	0.458	1,714
	... 11 additional rows in the appendix ...					

the filter does not classify them as scaffolding. Hyphae's residual `unsup_f` of 0.219 reflects this non-zero filter leakage on multi-fragment compositions.

GPT-4.1, on the same corpus, tightens its response under strong retrieval. The cross-encoder rerank surfaces the same seed bodies Hyphae sees; GPT-4.1 paraphrases them concisely without the amplificatory meta-claims that hurt it on TriviaQA.

5.4 The Latency-Quality Pareto Frontier

Figure 1 plots the 19 systems on three panels. The top row uses `unsup_f` on the vertical axis — the v1 preprint's headline grounding metric. The bottom-left panel uses gold-answer match instead, the correctness axis introduced in Section 5.1. We include both because `unsup_f` on TriviaQA is degenerate for Hyphae (rate 0.000 by quotient when the denominator collapses; the panel is illustrative, not load-bearing) and gold-answer is the axis that carries the architectural correctness claim.

The decisive feature of every panel is that Hyphae does not sit alone at the favourable corner: **the echo baseline is co-located with it** (and on the own corpus strictly dominates it, sitting at lower `unsup_f` and equal latency). Any "Hyphae is on the Pareto frontier" claim is therefore shared with a `print` statement, and the frontier on these axes is not evidence for the architecture. On the gold-answer axis the TriviaQA frontier also includes a handful of oracle-mode LLMs (Claude-4.6-Sonnet at 0.960, Llama-3.3-70B and GPT-4.1 at 0.900–0.907) at the high-correctness/high-latency end. No system dominates Hyphae on either axis on either corpus — but neither does Hyphae dominate echo, which is the point.

5.5 What Drives the Latency Ratio

The headline ratio varies by an order of magnitude across our measurements, and it is worth being precise about why. Table 4 reports mean per-query latency, held at fixed corpus, for Hyphae and representative LLM configurations. Two facts emerge that correct a tempting but wrong explanation.

Table 4: **Mean per-query latency, held at fixed corpus.** The ratio to HYPHAE is computed within each corpus block. HYPHAE's latency scales with working-set size (0.002 ms single-seed on TriviaQA, 0.024 ms multi-fragment on the own corpus); LLM latency is roughly corpus-independent. The network round-trip (local Llama vs API GPT-4.1 on TriviaQA) accounts for $\sim 30\%$, not an order of magnitude. Hyphae $p50/p95$ are omitted: at microsecond scale they sit at the timer-resolution floor and are not meaningful (the paper does not claim sub-microsecond precision).

System (mean latency)	mean (ms)	ratio to HYPHAE
TriviaQA-150 (single-seed queries)		
HYPHAE	0.002	1×
Llama-8B-Instruct Q4 local, oracle (MPS)	1,858	9.1×10^5
GPT-4.1 strong-rag (API)	2,424	1.2×10^6
Claude-4.6-Sonnet strong-rag (API)	5,370	2.7×10^6
Llama-3.3-70B strong-rag (API)	6,083	3.0×10^6
Own corpus (N=34, multi-fragment composition)		
HYPHAE	0.024	1×
Llama-8B-Instruct Q4 local, oracle (MPS)	2,299	9.4×10^4
GPT-4.1 strong-rag (API)	2,513	1.0×10^5
Hyphae on CPU droplet (Xeon, own corpus)		
HYPHAE	0.007	—

The network round-trip is not the dominant variable. On TriviaQA, the locally-hosted Llama-8B-Instruct Q4 baseline (no network; `llama.cpp` on the laptop's Metal backend) runs at 1,858 ms mean, and the API-hosted GPT-4.1 with strong retrieval runs at 2,424 ms mean. These differ by roughly 30% — the network round-trip is real but small. Both are $\sim 10^6\times$ slower than HYPHAE's 0.002 ms on the same corpus.

The dominant variable is Hyphae's working-set-dependent composition cost. On the own corpus, whose queries carry multi-fragment working sets (2–3 seeds), HYPHAE's mean latency is 0.024 ms — about 12× its single-seed TriviaQA latency, because the realizer walks a longer composition shape and runs more boundary-smoothing comparisons. The LLM latencies are roughly corpus-independent (their inference cost is dominated by model size and output length, not working-set count). The ratio therefore drops from $\sim 10^6$ on TriviaQA to $\sim 10^5$ on the own corpus. The order-of-magnitude difference between the two regimes is *HYPHAE getting slower as the working set grows*, not the LLM getting faster, and not the network.

The honest framing is therefore a range: the per-query latency advantage is $\sim 10^5$ for the multi-fragment composition workload and $\sim 10^6$ for single-seed retrieval, in both cases whether the LLM runs locally or via API. The title's "six orders of magnitude" refers to the single-seed regime; the conservative figure across both corpora is five orders. In either regime the HYPHAE side carries no model-serving infrastructure cost.

5.6 Hardware Matrix

Table 5 reports the laptop-vs-droplet comparison on the own corpus. The headline observation:

- **Quality metrics are hardware-invariant.** `verbatim_pass`, connective hygiene, quoted-content support, and n-gram overlap deltas are all within ±0.005 across the two hardware configurations. `unsup_f` drifts by ±0.01–0.03 (NLI floating-point noise across CPU vs MPS code paths). The architectural commitment is robust to hardware change.

Table 5: **Hardware matrix** on the own corpus. Quality metrics are hardware-invariant within ±0.03. Latencies are means; HYPHAE runs faster on the dedicated Xeon, the local Llama-8B baseline runs slower without the laptop's Metal backend.

	Hyphae		Llama-8B-local rag	
Metric	Laptop	Droplet	Laptop	Droplet
`verbatim_pass`	1.000	1.000	0.147	0.176
`unsup_f`(filt.)	0.219	0.188	0.490	0.476
overlap_4	0.466	0.466	0.448	0.449
latency mean (ms)	0.024	0.007	4,658	6,326

- **Hyphae is faster on the server CPU.** Mean latency 0.024 ms (laptop) → 0.007 ms (droplet). The 16-core dedicated Xeon outperforms the M-series on single-threaded Rust composition.
- **LLM is slower on the server CPU.** Without Metal acceleration the local Llama-8B rag path bottlenecks: 4,658 ms mean (laptop) → 6,326 ms mean (droplet); the oracle mode shows the same direction (2,299 → 4,580 ms). Because HYPHAE simultaneously *speeds up* on the dedicated Xeon (0.024 → 0.007 ms mean), the HYPHAE:Llama-rag latency ratio widens from $\sim 1.9 \times 10^5$ on the laptop to $\sim 9.3 \times 10^5$ on the droplet — i.e. removing GPU acceleration moves the advantage from the low-10^5 range toward 10^6.

5.7 Per-Model Behavioural Observations

Three patterns surface across the multi-LLM matrix that bear on how reviewers should interpret `unsup_f`, one of the two head-to-head metrics:

Claude-4.6-Sonnet's hedging. Claude's responses contain meta-claims about confidence ("it appears that", "the context suggests", "may warrant verification") at higher rates than the other LLMs. The NLI scorer marks these `neutral` against the context. Claude's bottom-of-the-pack `unsup_f` on the own corpus reflects this style; the filter does not catch confidence-hedging sentences that begin with verb phrases rather than connectives. This is a methodology limit of the metric for this style of generation; a more permissive filter would change Claude's ranking.

Llama-3.3-70B's regression versus Llama-8B-local. Llama-3.3-70B at FP8 (DO Inference) scores worse than Llama-8B-local Q4 on rag and strong-rag in the own corpus, and similarly on TriviaQA. Bigger model class is not a free lunch for this metric on these corpora. Hypotheses include FP8 quantisation differences and hosted-system prompt overhead (measured at ∼28 tokens versus ∼13 for other hosted models on trivial prompts).

GPT-4.1's behavioural verbatim-discipline. GPT-4.1 sits on or near the top of the LLM ranking in every configuration. Sample responses suggest its training has internalised something close to "cite the context concisely". The architectural commitment HYPHAE makes by construction, GPT-4.1 approximates by training.

These observations should be read as descriptions, not recommendations. The metric is one signal among many a production team would weigh.

6 Ablations

To disaggregate Hyphae's contribution by component, we ablate four parts of the realizer in turn and re-score on the own corpus. Each ablation disables a single component while leaving the other three (and the substrate's verbatim-quotation contract) intact.

Conditions.

- **A0 (full)**: control. All components active.
- **A1 (no-shape)**: cascade-shape composition (ADR-0006) disabled. The realizer is forced to a flat linear Continuation shape; opposed-valence Contrast detection and Causation/Sequence shapes are bypassed.
- **A2 (no-ethics)**: the ethics report at the Compose coverage point (ADR-0003) is set to `None`. The `EthicallySensitive` limitation trigger cannot fire. The other limitation triggers (`EmptyWorkingSet`, `HighConfabRisk`, `ShallowCascade`) still fire because they evaluate the working set directly.
- **A3 (minimal-lexicon)**: the baseline 250-entry EN lexicon (ADR-0005) is replaced with a 10-entry minimal lexicon (one entry per ConnectiveRole at Neutral register / Neutral polarity / Mid formality). The picker's four-level fallback chain resolves register and polarity preferences against the single minimal entry.
- **A4 (no-smoothing)**: boundary smoothing (ADR-0007) disabled. The realizer falls through to the plain context picker without the Rule 1/Rule 3 boundary filter.

Headline finding, read through the echo control. Table 6 shows the results. The architectural quotation contract (`verbatim_pass`, connective hygiene, quoted-content support) holds at 1.000 across every ablation, and gold-answer match is invariant (every condition still quotes the gold-bearing seed verbatim). Of the two head-to-head metrics, `unsup_f` moves measurably under **A3 (minimal lexicon) only**; A1, A2, and A4 produce null deltas.

We are careful about how much weight these ablations carry. The clean, load-bearing evidence for the reframing is the echo control (Section 5.1): it is a single, unambiguous comparison showing the correctness metrics are saturated by verbatim quotation. The ablations are *consistent* with that — if echo already saturates the metrics, ablating the machinery layered on top should not move them, and it does not — but at $N = 34$ they are underpowered, so we present them as corroboration, not as independent proof. Two honesty notes. (i) A2 (no-ethics) is a *declared no-op* on this corpus, not a real ablation: the corpus contains no ethically-sensitive material, so the `EthicallySensitive` trigger had nothing to fire whether or not the ethics report was supplied; its null delta is definitional. (ii) A3 (minimal lexicon) does move `unsup_f` slightly, but in inconsistent directions across the filtered and unfiltered variants (Table 6), so we do not read it as a clean effect either. What the ablations *do* establish without caveat is orthogonal to correctness: **every component, and both the echo and echo+journal baselines, leaves the hash-chained provenance relation intact** — no configuration in this paper can produce an output whose quoted spans are not byte-identical to journalled fragments. The provenance layer is the one axis nothing here perturbs, and (per Section 5.2) it is not Hyphae-specific.

Table 6: **Ablation results on the own corpus.** Quality contracts are robust to each single-component ablation. Only A3 (minimal lexicon) measurably moves `unsup_f` and overlap_4. **Bold** = delta of ≥ 5 percentage points versus the A0 baseline.

Metric	A0 full	A1 no-shape	A2 no-ethics	A3 min-lex	A4 no-smooth
`verbatim_pass`	1.000	1.000	1.000	1.000	1.000
connective hygiene	1.000	1.000	1.000	1.000	1.000
quoted-content supp.	1.000	1.000	1.000	1.000	1.000
overlap_4	0.466	0.460	0.466	**0.521**	0.457
overlap_8	0.240	0.236	0.240	0.272	0.234
`unsup_f`(filt.)	0.219	0.188	0.219	0.297	0.188
`unsup_r`(raw)	0.625	0.656	0.625	**0.461**	0.641
latency mean (ms)	0.024	0.056	0.048	0.030	0.021

6.1 Mechanism Analysis

A1, A2, A4: null deltas with visible structural change. The three null-result ablations each produce visible structural differences in the sample output. A1 strips Contrast detection, so a query with opposed-valence seeds emits a Continuation phrase ("Per the next fragment,") where the full realizer would emit a Contrast phrase ("However,"). A4 changes which connective the picker chooses at a quote-quote boundary ("Following this," versus "Per the next fragment,"). A2 strips the `EthicallySensitive` signal, but the v0.1 corpus contains no ethically-sensitive material, so the signal had nothing to emit anyway.

The metric set does not separate these structural changes from the baseline. This is informative in itself: at the corpus size and metric set chosen, three of Hyphae's components produce qualitative effects that do not register quantitatively. A production deployment that values prose variety or limitation-acknowledgement coverage would still want them enabled; the metric does not measure those properties.

A3: lexicon scale moves both directions of the metric. The minimal lexicon raises overlap_4 from 0.466 to 0.521 (+12%) because its phrases ("Note that," "Per the record:") are shorter and contribute fewer non-context tokens to the overlap denominator. The same shortening cuts the `unsup_r` rate from 0.625 to 0.461 (-26%): there are fewer scaffolding sentences for the NLI to mark `neutral`.

The `unsup_f` rate moves the other direction: $0.219 \rightarrow 0.297$. The shorter minimal-lexicon phrases sometimes slip through the `is_connective_sentence` filter (e.g., "Also," and "Specifically," are not in the filter's keyword list); the filter under-counts and lets through scaffolding that the NLI then marks `neutral` as if it were a claim. The `unsup_f` rise in A3 reflects this filter gap, not a fidelity defect.

Causal attribution. A3's directionally-correct deltas explain a phenomenon visible in the head-to-head: Hyphae's overlap_8 on the own corpus (0.240) is *lower* than the LLM baselines' overlap_8 (0.329). The mechanism is now identified: the baseline EN lexicon's longer phrases break 8-token windows around each verbatim quote. The minimal lexicon raises overlap_8 from 0.240 to 0.272 — partial closure of the gap, attributable to lexicon scale rather than to fidelity. A shorter lexicon tuned for n-gram-friendly phrases could close the gap further; that is a separate ADR.

6.2 Limits of the Ablation

We ran four single-component ablations, not a $2^4 = 16$ factorial. At $N = 34$ the statistical separation of single-component effects is already strained; full factorial would compound the multiple-comparisons problem. We do not claim that pairwise interactions are negligible — only that the single-component sweep is what the corpus size supports.

We did not ablate the verbatim-quotation contract itself (Hard Commitment 12). Ablating that commitment converts the system into a paraphrase pipeline — in effect, an LLM. The LLM-baseline columns of the head-to-head are exactly that ablation, run on six different generators. The result is in Section 5; we do not double-count it as an ablation here.

7 Discussion

7.1 Threats to Validity

The metrics cannot measure the contribution (the central threat). The echo baseline establishes this directly: a verbatim `print` of the retrieved sentence ties or beats Hyphae on gold-answer match, `unsup_f`, `unsup_r`, and n-gram overlap on both corpora. Every correctness and grounding number in this paper is therefore a measurement of verbatim quotation, a property echo shares, and not of Hyphae's architecture. We do not treat this as a flaw in the experiments but as the finding: the standard correctness benchmark is the wrong instrument for a verifiable-provenance system, because the quantity that distinguishes such a system — whether the output is tamper-evident against a store-level adversary, auditable back to named source fragments (Section 5.2) — is off-axis for it. `unsup_f` additionally penalises hedging (Claude-4.6-Sonnet) and rewards short verbatim responses, and we report `unsup_r` (unfiltered) alongside it so the filter's effect is visible; but no amount of metric refinement closes the gap the echo control exposes, because the gap is categorical, not quantitative. A defensible evaluation of Hyphae requires a provenance-centric task (Section 7, future work), not a better-tuned correctness metric.

Corpus. The own corpus is concentrated on deployment and engineering scenarios and was authored by the same team that designed Hyphae's schemas and intents; empirical ties on that corpus should be read with that partial circularity in mind. TriviaQA addresses the standard-benchmark objection but is single-hop factual retrieval over Wikipedia and was sampled with a filter that selects seed bodies containing the gold answer. Hyphae's gold-answer rate on this corpus is therefore 1.000 by construction. Multi-hop benchmarks (HotpotQA, MuSiQue), domain-specific benchmarks (TruthfulQA, MS MARCO), and long-context benchmarks (NarrativeQA) would each surface different patterns and are reasonable next directions.

Statistical power. $N = 34$ supports the patterns we emphasise (Pareto frontier shape, latency advantage, Hyphae-vs-GPT-4.1 tie on `unsup_f`) but admits multiple rank-order swaps in the middle of the ranking. $N = 150$ substantially tightens the TriviaQA picture: the rank-1 position of Hyphae is robust to bootstrap resampling, but middle-of-the- table positions remain noisy.

Reader preference. Hyphae's prose is template-rigid. The realizer's output reads as deliberate rather than conversational; this is by design and motivated in Section 3. Whether deliberate prose is preferred by end users on factual-retrieval queries is a separate study (human eval) that this paper does not provide. The honest claim is that Hyphae's prose meets the verbatim-grounding bar by construction and that an end-user study is the natural next checkpoint.

API non-determinism. Closed-API generators (Claude-4.6-Sonnet, GPT-4.1) do not guarantee bit-identical reruns under `temperature=0, seed=42`. Per-call variation is small at greedy decoding but non-zero; our reported results are a snapshot. The result JSON envelopes are committed; reviewers can re-run and inspect any drift. Open-weight models on the same OpenAI-compatible endpoint are substantially more deterministic across reruns.

Hardware sensitivity. The hardware matrix (Section 5.6) tests two configurations and finds quality metrics invariant within ±0.03. A wider matrix (GPU server class, different x86 microarchitectures, ARM server) is queued; the v0.1 paper does not generalise across all hardware. The latency advantage we report on the laptop and the CPU droplet, however, holds at both points; the LLM never reaches sub-millisecond latency in our measurements.

7.2 When This Matters

The architectural claim — five-to-six orders of magnitude lower latency at competitive grounding — pays off most clearly in three regimes.

High-throughput grounded retrieval. A service whose fundamental operation is "cite the relevant memory fragment for this query" bottlenecks on LLM serial inference today. HYPHAE's *composition* stage removes that bottleneck: the 7 μs realizer latency on the droplet implies $\sim$ 143,000 *compositions* per second per core. This is a composition- throughput figure, not an end-to-end system figure: it excludes ANN retrieval, the journal append, and the SHA-256 hashing on the write path, which dominate a real deployment's per-query cost and which we have not throughput-profiled. Even with those included, the relevant comparison is qualitative and robust: an LLM serving the same task at $\sim$ 5,000 tokens/s on a high-end GPU fits on the order of single-digit queries per second per GPU, so the per-query cost differs by several orders of magnitude regardless of where exactly the substrate's end-to-end number lands.

Audit-bound deployments. The hash-chained substrate and verbatim quotation are direct architectural answers to "can you prove what this system told the user came from named source material?". Industries with audit requirements (finance, legal, compliance, healthcare) currently rely on LLM-output logging plus post-hoc verification. HYPHAE folds verification into the output by construction.

Memory-constrained edge deployments. The substrate's $\sim$ 50 MB resident footprint runs on commodity hardware without GPU. Production patterns that today require model serving infrastructure (Llama-8B in $\sim$ 5 GB, Llama-70B in $\sim$ 40+ GB) cannot fit; HYPHAE fits.

7.3 When This Does Not Matter

We do not claim the architectural commitment is universally correct. Three regimes call for an LLM.

Open-ended generation. If the user wants the system to write prose — a draft letter, a summary in their voice, a creative composition — the verbatim-quotation commitment forbids the operation. The right tool is an LLM, optionally augmented by HYPHAE as a memory layer.

Multi-hop reasoning. HYPHAE composes verbatim quotes; it does not infer across multiple fragments. A query that requires synthesising "X was born in 1822" and "the Y revolution began in 1830" to answer "how old was X at the time of Y?" needs a reasoning step the realizer does not perform. An LLM, or HYPHAE-plus-LLM where the LLM operates on HYPHAE's retrieved fragments, is the right tool.

Paraphrastic flexibility. HYPHAE's prose reads as template prose. If conversational naturalness is a hard requirement, the architecture's trade-off is in the wrong direction.

The honest framing is positional: HYPHAE occupies a specific point in the latency-quality-cost-audit space. The paper documents that point.

7.4 Future Work

Scaling the provenance benchmark. The tamper-detection experiment (Section 5.2) demonstrates the property in the small, on a single hash chain, with the chain-aware adversary closed by an implemented Ed25519 head anchor (ADR-0032). A community-scale provenance benchmark — diverse tampering models, adversaries with varying store and key access, and a standard detection-and-localisation scoring protocol — would let verifiable-generation systems be compared on the axis that distinguishes them, the way correctness benchmarks compare LLM-RAG systems today. Building it is the natural next contribution.

Logarithmic verification. Our journal is a *flat* hash chain: verifying the whole log is $O(n)$. Certificate Transparency [Laurie et al., 2013] and its instantiations achieve $O(\log n)$ inclusion and consistency proofs with a Merkle tree over the same append-only semantics. At the per-fragment granularity and audit cadence we target the flat chain is a deliberate simplification, not a limitation of the property; a Merkle upgrade is a drop-in for deployments that need to prove inclusion or append-only consistency to a verifier without streaming the entire log, and it does not change any result in this paper. We make this concrete: the provenance benchmark (`hyphae-provbench`) carries a Merkle/RFC-6962 transparency log as a comparator system alongside the flat chain. The two *match* on every detection cell — confirming that detection is a property of the append-only-log class, not of the flat chain specifically — and differ only on inclusion-proof cost ($O(n)$ versus $O(\log n)$), which the benchmark reports as a per-system axis. A no-chain per-entry-signature log, included as a third comparator, catches in-place edits but misses deletion, reordering, replay, and rollback, separating "signing" from "chaining."

External anchor publication and key management. The head anchor we implement signs the head with a key held outside the store, which catches the chain-aware attacker. Publishing those anchors to an append-only, hash-chained ledger — so that *freshness* (a rolled-back head is rejected even when the attacker replays its genuine but superseded anchor) and *non-equivocation* (forked views are detected) hold across snapshots, not just integrity of the latest head — is implemented in the repository as the `hyphae-storage::ledger` module (ADR-0033), with an adversarial demonstration. The remaining withholding attack — a store that presents a truncated ledger as the latest — is closed by an external *witness*: an independent party (a separate key) that observes and signs the ledger tail, against which the auditor pins how far the ledger really went (`hyphae-storage::witness`, ADR-0034). What remains is deployment engineering, not open research: binding the witness to a concrete external service (a timestamp authority, transparency-log witness, or OpenTimestamps/Bitcoin commitment of the ledger head). Key rotation itself — so a compromised signing key is recoverable rather than fatal — is implemented as a signed keyring (`hyphae-storage::keyring`, ADR-0035): successors are authorised by their predecessor from a root trusted out-of-band, and a ledger spanning rotations verifies under the key active at each epoch; what is left is KMS/HSM sourcing of the key material. The source-ingestion boundary — attesting that fragments entered the journal faithfully in the first place — is addressed at the protocol level by an *ingestion bridge* (`hyphae-storage::ingestion`, ADR-0037): at admission an ingestion source signs a credential, aligned to C2PA Content Credentials, binding a fragment's exact

bytes to a claimed origin document and byte-range locator, and journaled as a typed event so it inherits the integrity chain. This makes origin *attributable* (who asserted, from where) and, given the source, *faithful-excerpt verifiable* (the fragment is byte-for-byte `source[locator]`), moving the trust boundary from the store to a named asserter. What it deliberately does not establish is the *truth* of the source itself: content validity — whether a faithfully-ingested fragment is correct, e.g. a plausible-but-wrong poisoned value — is an orthogonal axis left to complementary content-checking work.

Reader preference study. Human evaluation on a subset of the TriviaQA corpus. The template-rigid prose's acceptability to real users is unmeasured, and is the main usability question the verifiability framing does not resolve.

Multi-hop benchmark column. HotpotQA or MuSiQue subset. HYPHAE's single-fragment composition cannot synthesise across hops; the open question is whether it degrades gracefully (deferring to limitation acknowledgement) or silently. We are explicit that the *principle* of abstaining on insufficient support is not new: Menick et al. [2022] abstain on the most-uncertain queries in attributed QA, and unanswerable-question detection is a standard task since SQuAD 2.0 [Rajpurkar et al., 2018]. Our contribution here is narrow and measurement-shaped — an offline harness quantifying that a single-span verbatim realizer *silently* fails on multi-hop unless an abstention signal is made explicit — not a new abstention method.

Hardware and retrieval extensions. A GPU server-class matrix would close the LLM's biggest performance handicap; stronger retrieval pipelines (RAG-Fusion, GraphRAG) on the LLM side could shift the correctness comparison. Neither changes the provenance result, which is a property of the storage layer, not the retrieval quality.

8 Conclusion

We have presented HYPHAE, a cognitive substrate that composes responses by verbatim quotation of retrieved memory fragments stitched together with a curated connective lexicon and produces an audit relation to a hash-chained substrate journal. Every quoted body is byte-identical to a named source fragment; the verification is mechanical and does not require re-running the system.

Our central empirical result is a negative one, and we lead with it. A trivial echo baseline — emit the retrieved sentence verbatim, no model — ties or exceeds HYPHAE on every correctness and grounding metric we measured, on both corpora. The standard benchmark cannot distinguish HYPHAE from a `print` statement, and our ablations confirm why: the metrics are saturated by verbatim quotation alone, so the composition machinery layered on top moves them only marginally or not at all. We therefore explicitly decline the claim that HYPHAE is more correct or better-grounded than LLM-RAG. On correctness it is *neutral* — it does not lose gold-answer accuracy relative to echo or to frontier LLMs in oracle mode (0.840–0.960) — and that is the honest ceiling of what the benchmark establishes.

What the benchmark cannot see is the property that motivated the architecture. Echo emits text with no relation to a tamper-evident store; HYPHAE emits spans that are byte-identical to fragments in a SHA-256 hash-chained journal, so any output can be mechanically audited back to named, unaltered sources without re-running the system. No retrieval-augmented LLM offers this by construction, because generation paraphrases; no plain echo offers it either, because it has no journal — but echo+journal does, which is why we frame the contribution as an addable layer

rather than a property of HYPHAE. A chain-aware attacker who rewrites the persisted head defeats the bare chain; an implemented external head anchor (Ed25519, key held outside the store) closes that gap, demonstrated in §5.2. This layer is the contribution, and it is precisely the axis on which every metric in the standard correctness benchmark is silent.

The cost profile is a corollary of the same decision not to run a generative model: HYPHAE's realizer runs in 2–24 microseconds mean on commodity CPU, five-to-six orders of magnitude below LLM inference, in a ~ 50 MB footprint. This is real and matters for deployment, but it is shared with the echo baseline and is not, on its own, evidence for the architecture either. For the subset of tasks in which the unit of grounding is a discrete retrieved fragment quoted verbatim *and* the deployment requires provable provenance, the trade-off is not between two systems of comparable cost; it is between a single CPU core that can prove what it emitted and model-serving infrastructure that cannot.

The architectural claim is positional rather than universal. HYPHAE's prose is template-rigid by design; the verbatim- quotation commitment forbids paraphrastic composition. Applications requiring fluent free-form generation need an LLM. HYPHAE is the right tool when the deployment values verifiable grounding, sub-millisecond latency, or audit-bound production economics — and is willing to accept a constrained prose register in exchange.

The full evaluation pipeline (HYPHAE source, the echo control, the 18 LLM configurations, two corpora, all result envelopes, hardware traces, and this paper's source) is committed at `https://github.com/terrizoaguimor/hyphae-v2`. Reproducing the experimental matrix costs approximately USD 4 in inference tokens at the LLM API layer; the HYPHAE and echo sides have zero cloud cost. The echo baseline tells us the most important next step: the standard correctness benchmark is the wrong instrument for this system, because a `print` statement saturates it. The honest next checkpoint is therefore not a larger correctness corpus but a *provenance-centric evaluation* — a task in which verifiable audit of the output's sources is the measured desideratum (regulatory citation, compliance logging, tamper-evident record-keeping), where echo and LLM-RAG both structurally fail to provide the guarantee and HYPHAE provides it by construction. Building that benchmark, and a reader-preference study of the template-rigid prose, are the two followups that would let the architecture be evaluated on the axis that actually distinguishes it.

Acknowledgements

This work was developed by Celiums Solutions LLC, Medellín, Colombia. Compute for the multi-LLM matrix was provided by DigitalOcean's GenAI Platform under the standard infrastructure account associated with the project's other services. No academic or industrial collaborators participated in the research or writing.

Data and Code Availability

All code, corpora, result envelopes, and this paper's source are available under a dual licensing scheme at `https://github.com/terrizoaguimor/hyphae-v2`: code under Apache License 2.0, documentation and corpora under Creative Commons Attribution 4.0 International.

References

John R Anderson. *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press, 2007.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *ICLR*, 2024.

BAAI. BGE-Reranker: cross-encoder reranking models from BAAI, 2024. https://huggingface.co/BAAI/bge-reranker-base.

Celiums Solutions. ADR-0003: Ethics-RADAR first-class, 2026a. https://github.com/terrizoaguimor/hyphae-v2/blob/main/docs/adr/0003-ethics-radar-firstclass.md.

Celiums Solutions. ADR-0005: Lexicon scale, 2026b. https://github.com/terrizoaguimor/hyphae-v2/blob/main/docs/adr/0005-lexicon-scale.md.

Celiums Solutions. ADR-0006: Cascade-shape-driven composition, 2026c. https://github.com/terrizoaguimor/hyphae-v2/blob/main/docs/adr/0006-cascade-shape-driven-composition.md.

Celiums Solutions. ADR-0007: Boundary smoothing, 2026d. https://github.com/terrizoaguimor/hyphae-v2/blob/main/docs/adr/0007-boundary-smoothing.md.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating LLM inference with staged speculative decoding. *arXiv preprint arXiv:2308.04623*, 2023.

DigitalOcean. Digitalocean GenAI platform: OpenAI-compatible inference for open and frontier models, 2025. https://www.digitalocean.com/products/gen-ai.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A GraphRAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*, 2022.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*, 2023.

Zorik Gekhman, Jonathan Herzig, Roee Aharoni, Chen Elkind, and Idan Szpektor. TrueTeacher: Learning factual consistency evaluation with large language models. *arXiv preprint arXiv:2305.11171*, 2023.

Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3(2):99–111, 1991.

Christopher Hesterberg. redb: A simple, portable, high-performance, ACID, embedded key-value store, 2024. https://github.com/cberner/redb.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansky, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. TRUE: Re-evaluating factual consistency evaluation. In *NAACL*, 2022.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023.

Marvin Jablonski. fjall: LSM-tree based embedded key-value database, 2024. https://github.com/fjall-rs/fjall.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 2023.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, 2017.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP*, 2020.

John E Laird. *The Soar Cognitive Architecture*. MIT Press, 2012.

Ben Laurie, Adam Langley, and Emilia Kasper. Certificate transparency, 2013. RFC 6962, IETF.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *ICML*, 2023.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*, 2020.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Yury A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2018.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*, 2022.

Ralph C Merkle. A digital signature based on a conventional encryption function. In *CRYPTO '87*, pages 369–378. Springer, 1988.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*, 2019.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2024.

Zackary Rackauckas. RAG-Fusion: A new take on retrieval-augmented generation. *arXiv preprint arXiv:2402.03367*, 2024.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822*, 2018.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP*, 2019.

Tal Schuster, Adam D. Lelkes, Haitian Sun, Jai Gupta, Jonathan Berant, William W. Cohen, and Donald Metzler. SEMQA: Semi-extractive multi-source question answering. *arXiv preprint arXiv:2311.04886*, 2023.

Linus Torvalds, Junio Hamano, et al. Git: a distributed version control system (content-addressable object store), 2005. https://git-scm.com.

Jonas Wallat, Maria Heuss, Maarten de Rijke, and Avishek Anand. Correctness is not faithfulness in RAG attributions. *arXiv preprint arXiv:2412.18004*, 2024.

Orion Weller, Marc Marone, Nathaniel Weir, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. "according to ...": Prompting language models improves quoting from pre-training data. *arXiv preprint arXiv:2305.13252*, 2023.

Theodora Worledge, Tatsunori Hashimoto, and Carlos Guestrin. The extractive-abstractive spectrum: Uncovering verifiability trade-offs in LLM generations. *arXiv preprint arXiv:2411.17375*, 2024.

A Full Per-System Tables

The complete 19-system tables for both corpora are available as machine-readable JSON envelopes in `bench/baseline-llm-rag/results/` in the repository. Each envelope carries the per-query trace, metric breakdown, bootstrap 95% percentile confidence intervals (1000 resamples, seed 42), hardware metadata, and model identifier. The Section 5 tables are sorted versions of those envelopes' aggregates and a future revision of this paper will expand them inline.

B Reproduction Instructions

B.1 Environment Setup

```
git clone https://github.com/terrizoaguimor/hyphae-v2
cd hyphae-v2

# Rust toolchain (1.85+)
rustup default stable

# Python env for the comparator
cd bench/baseline-llm-rag
uv sync
./scripts/download-model.sh  # Llama-8B Q4_K_M (~5GB)
cd ../..
```

Listing 3: Setup commands.

B.2 Corpus Generation

```
# Own corpus -> JSON
cargo run --quiet -p hyphae-eval --example export_corpus \
    > bench/baseline-llm-rag/corpus-en.json

# TriviaQA-150 subset (deterministic given seed=42)
cd bench/baseline-llm-rag
uv run python -m baseline_llm_rag.corpus_external \
    --seed 42 --n 150 --output corpus-triviaqa-150.json
```

Listing 4: Export the own corpus and build the TriviaQA-150 column.

B.3 Run the Matrix

```
export DO_INFERENCE_KEY='<your-token>'

# Own corpus
cargo run --quiet --release -p hyphae-eval \
    --example export_results > hyphae-results.json
uv run python -m baseline_llm_rag.score_hyphae \
    --hyphae-output hyphae-results.json \
    --output results/v0.1-laptop-hyphae-none.json
for mode in oracle rag strong-rag; do
    uv run baseline-llm-rag --mode $mode \
        --corpus corpus-en.json \
        --output "results/v0.1-laptop-${mode}.json"
done
for model in llama3.3-70b-instruct anthropic-claude-4.6-sonnet \
             openai-gpt-4.1 deepseek-v4-pro \
             router:celiums-conversation; do
    tag=$(echo "$model" | tr ':' '-')
    for mode in oracle rag strong-rag; do
        uv run baseline-llm-rag --mode $mode \
            --corpus corpus-en.json \
            --output "results/v0.1-doinf-${tag}-${mode}.json" \
            --llm-backend do-inference --model "$model"
    done
done

# TriviaQA: same shape, --corpus corpus-triviaqa-150.json
bash scripts/run-triviaqa.sh
```

Listing 5: Run Hyphae, local Llama-8B, and the DO Inference 5-model matrix on both corpora.

B.4 Ablations

```
for ablation in none no-shape no-ethics minimal-lexicon \
                no-smoothing; do
    cargo run --quiet --release -p hyphae-eval \
        --example export_results_ablation \
        -- --ablation "$ablation" \
        > "bench/baseline-llm-rag/hyphae-results-${ablation}.json"
    cd bench/baseline-llm-rag
    uv run python -m baseline_llm_rag.score_hyphae \
```

```
        --hyphae-output "hyphae-results-${ablation}.json" \
        --output "results/v0.1-laptop-hyphae-${ablation}.json"
    cd ../..
done
```

Listing 6: Run the four ablation conditions.

C Selected Sample Outputs

We include sample outputs from selected systems on selected queries to give the reader concrete texture for the numbers in Section 5. The full per-query output is in the JSON result envelopes in `bench/baseline-llm-rag/results/`.

Own corpus, `dialogue-001` (*"what is the status of the migration?"*)

Hyphae: *"Per the recorded fragments, 'the migration completed at 14:02 UTC' Per the next fragment, 'the monitoring dashboards stayed green for the hour after the cutover' That is the substrate's current view."*

GPT-4.1 strong-rag: *"The status of the migration is that it completed at 14:02 UTC, with monitoring dashboards staying green for the hour after the cutover."*

TriviaQA, `triviaqa-qb_4397` (*"In April 1963, Lester Bowles Pearson became Prime Minister of which country?"*)

Hyphae: *"Drawing from working memory, 'Pearson did not make the move into politics until a few years later, after King had announced his retirement as the Prime Minister of Canada.' That is what working memory holds on this."*

Llama-3.3-70B rag: *"According to the provided context, Lester Bowles Pearson became Prime Minister of Canada in April 1963. This information is supported by the sentence which mentions that Pearson did not enter politics until after King had announced his retirement as the Prime Minister of Canada, indicating that Pearson eventually took this role himself."*

The Llama-3.3-70B response contains three sentences. The factual content (Pearson, Canada) is correct. The first sentence ("According to the provided context, Lester Bowles Pearson became Prime Minister of Canada in April 1963") is scored `neutral` by the NLI: the context does not explicitly state "April 1963", although the answer is correct in the world. The second sentence describes how the answer relates to the context; `neutral`. The third sentence infers an action ("Pearson eventually took this role himself") not in the context; `neutral`. All three count as unsupported claims by the filtered metric.

This is a sample of the mechanism by which the unsupported-claim rate accumulates on LLM responses to TriviaQA queries even when the underlying factual answer is correct.

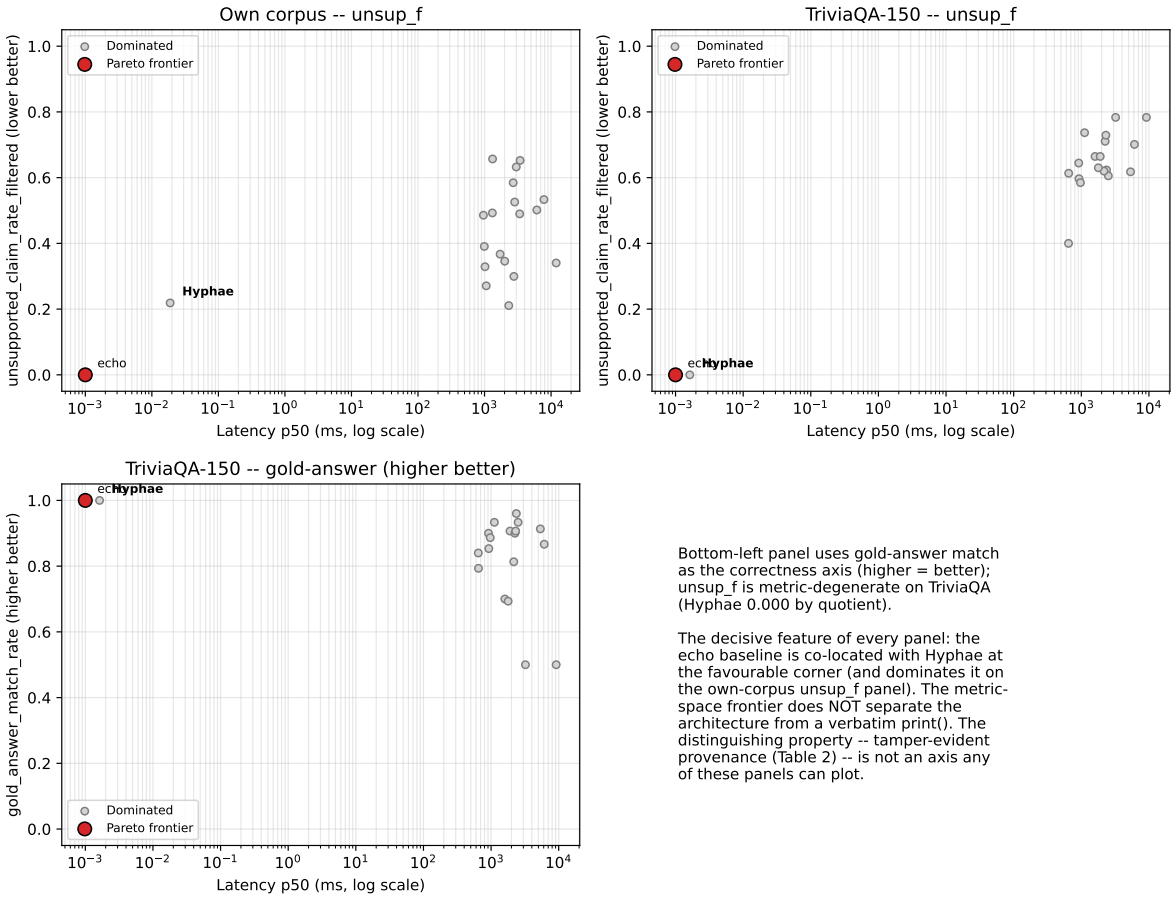

Figure 1: **Latency-quality frontier on two metrics, with the echo control.** Each point is one system configuration; latency on the horizontal axis is $p50$ in milliseconds, log scale. The top row uses `unsup_f` and is *illustrative only*: `unsup_f` is metric-degenerate for verbatim systems (Hyphae and echo both reach 0.000 on TriviaQA by quotient; §5.1), so the load-bearing correctness axis is the bottom-left panel, gold-answer match (higher better). Red points are non-dominated. On every panel the echo baseline sits at the same favourable corner as Hyphae (and dominates it on the own-corpus `unsup_f` panel): the metric-space frontier does not separate the architecture from a verbatim `print`. The distinguishing property — tamper-evident provenance, measured in Table 2 — is not an axis any of these panels can plot.