Both Engines, One Harness: A Self-Run Head-to-Head Against mem0's Open-Source Server

Naïve Research

Naïve, Inc. · team@usenaive.ai · July 2026

Abstract

Every published number in agent memory has to be read against a broken backdrop. The most-cited long-conversation benchmark has an independently audited 6.4 percent wrong answer key, its automated judge accepts 62.8 percent of intentionally wrong answers, and one published system has been reproduced more than fifty points below its own reported score. When two parties' figures for one system disagree that far, the harness is deciding the score, not the engine. The only escape is to obtain the competitor's software and run both engines yourself. Run against mem0's own open-source server on an identical harness, with the same answerer and judge, Brain scored 88.0% to mem0's 66.6% on 778 held-out LoCoMo questions, including 78.8% against 12.7% on temporal reasoning. Three things travel with that number and are stated in full rather than footnoted: mem0's own harness container does not build, so we pinned `mem0ai==2.0.13` from PyPI, and this is not the v3 pipeline they benchmarked; LoCoMo's answer key is 6.4 percent wrong, so neither figure is precise to the decimal; and the Brain arm ran a pre-2026-07-25 configuration, which makes the result a lower bound rather than a measurement of the shipped system. We describe the engine on our side of the comparison: a diary and fact-sheet split, an append-only event log of raw episodes that is never mutated, plus a derived layer of bitemporal claims that are superseded rather than overwritten, with consolidation kept off the query path. On cost, against mem0's published configuration, 7,000 tokens per query at a top-200 retrieval budget, Brain answers from roughly 780 tokens of retrieved context at top-20, for about 2.4x lower cost per query, and the clause naming which mem0 is part of the figure rather than a qualification of it. An earlier version of this paper led with a growing-history read-cost result of our own and took its title from it. That result has been withdrawn to internal pending hardening actions on its measurement log that remain open, and section 10 says so.

---

1 No benchmark number survives a second harness

The agent-memory literature races on recall: which system answers the most questions correctly over a long conversation. The race is run on numbers that do not survive being re-run. LoCoMo, the most-cited long-conversation benchmark, has an independently audited 6.4 percent wrong answer key, which caps achievable accuracy near 93.6 percent, and its own automated judge accepts 62.8 percent of intentionally wrong answers. Third-party reruns disagree with vendor self-reports by margins wider than the gaps the vendors are claiming. The clearest case ran in public and in three acts: Zep published roughly 84 percent on LoCoMo, mem0's CTO re-ran Zep's own pipeline with the adversarial category handled consistently and got 58.44 percent, and Zep then conceded an arithmetic error and corrected its own figure to 75.14 percent. We state all three because the last one is the part a competitor would omit, and omitting it is the behaviour this section is complaining about. One system, one dataset, one year, and a spread no algorithm accounts for. A separate reproduction of EverMemOS put a published 92.32 percent at 38.38 percent. When the same system can score 38 or 92 depending on who runs the harness, the harness is deciding the score.

There is one escape and it is not a better leaderboard. Obtain the competitor's software, stand it up beside your own, and put both through the same harness with the same answerer, the same judge, the same retrieval cutoff, and the same machine, so that the engine is the only variable left. That is more work than citing a

number, and it is the only form of evidence in this field we think is worth much. This paper reports what happened when we did it against mem0's open-source server, describes the memory engine on our side of the comparison, and states what a query costs against the one mem0 configuration where we can state it honestly.

Accuracy is not the only axis, and for a fleet of long-lived agents it is probably not the binding one. A memory system exists because reading the whole history back is expensive, and that expense grows with the history. ConvoMem measures the curve directly: on its corpus a full-context reader's cost rises from about 0.001 to 0.09 dollars per query by 300 conversations while a retrieval-based memory system's stays roughly flat, and full context is nonetheless the more accurate option for the first 150 conversations or so, after which cost and context degradation flip the trade. Memory is a scale optimization, not an early accuracy feature. Section 7 states what a query costs on our side and against which mem0; section 10 states which of our own cost results are not currently citable, and why.

This paper has been rewritten and retitled. An earlier version led with a growing-history read-cost result of our own and took its title from it. That result has since been withdrawn to internal in our claims ledger, because hardening actions against the instrument that produced it are still open, and it does not appear here in any form. Section 10 sets out what went and why. A paper that quietly drops its thesis is harder to trust than one that says what it lost.

2 Architecture: a diary and a fact sheet

Brain is structured as two layers with different jobs. A **diary** is an append-only event log of raw episodes; nothing in the system ever mutates an episode. A **fact sheet** is a derived layer of bitemporal claims that are versioned and superseded rather than overwritten. Reads go to whichever layer serves the question; writes only ever append; and consolidation, the work of summarizing, merging, decaying, and evicting, runs offline, off the query path. Figure 1 shows the shape.

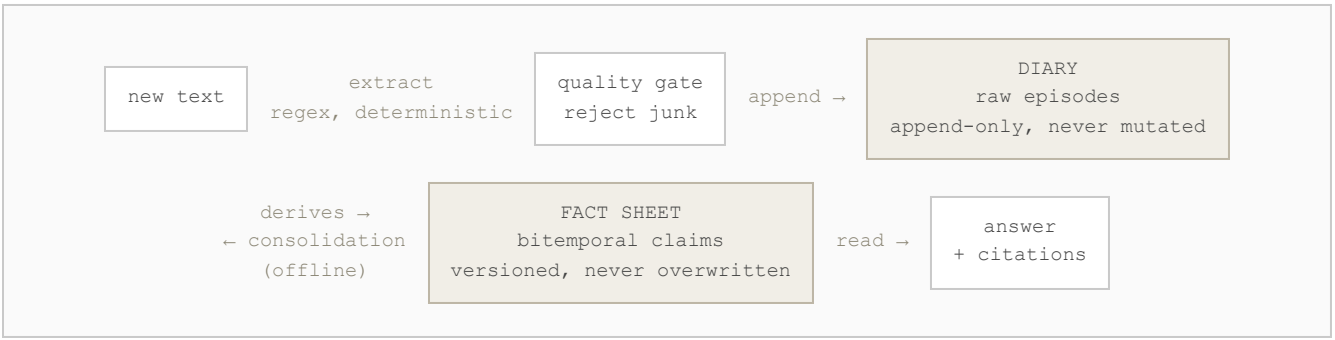

Figure 1. The diary and fact-sheet split. Writes only append; the fact sheet is a derived projection of the log; consolidation runs offline, so a read never pays for write-time work. Every belief traces back to the episode that produced it, and history is never overwritten.

2.1 The substrate is an append-only event log

Raw episodes are appended to an immutable log, and everything else, claims, summaries, graph edges, is a derived projection of it. Nothing in the system mutates an episode. This is event sourcing applied to memory, and the 2026 literature converged on it as an explicit cluster: The Log is the Agent (arXiv:2605.21997) treats an append-only event log as the source of truth with the working graph as a deterministic projection, naming a determinism contract; ESAA (arXiv:2602.23193) has agents emit schema-validated intentions that a

deterministic orchestrator applies to an append-only file, with the materialized view derived by replay and verified with a projection hash. A hash-verified projection is the closest published thing to a correctness proof for a derived memory layer, and it is the direction our replay testing points.

2.2 Bitemporal claims and functional supersession

A claim carries both valid time (when the fact was true in the world) and transaction time (when we recorded it). Updating a fact does not overwrite it; it closes the old claim's validity window and opens a new one, so a time scoped recall, `recall(as_of=T)`, reconstructs what we believed at any past instant T. Supersession instead of overwrite is cheap precisely because the log is already immutable. The only widely cited published implementation of bitemporal agent memory with invalidation is Zep/Graphiti (arXiv:2501.13956); we adopt its as-of-aware retrieval, searching the claim space as it stood at T, on top of storage that already predated it. We carry one caveat from that line deliberately: a third-party re-evaluation put Zep at 58.4% on a benchmark it published at 84%, which is exactly why we do not cite vendor numbers bare, including our own (section 8).

3 The write path: deterministic-first

Default ingest is deterministic, regex and rule based. An LLM distiller exists, is implemented, and is tested, but it is not the go-forward default. We are not stating a verdict here on whether it earns its cost. The instrument that could settle that question, a per-stage token counter on the write path, did not exist when the decision was taken, and the comparison has not been re-run since it did. What follows is why the deterministic default was a reasonable prior, and it comes from the published field rather than from us.

The literature explains why the write-time LLM is a conditional lever, not a free one. TierMem names the write-before-query barrier: write-time summarization makes irreversible retention decisions before the queries are known, and its example is unforgettable, a "severe peanut allergy" compressed into "dietary preferences." It measures summary-only baselines carrying a 15 to 30 percent unverifiable omission rate. General Agentic Memory frames extract-at-write as ahead-of-time compilation with inevitable loss, versus just-in-time retrieval over a preserved page store. The synthesis both papers reach, and ours, is ahead-of-time for hot typed state, just-in-time escalation for the long tail. And a controlled strategy comparison (arXiv:2604.01707) finds that explicit rule-based hierarchical management holds its accuracy as the memory scales, while LLM-driven management degrades into tool-call failures and indexing conflicts once the candidate space grows. The cost of the LLM-heavy path is large and measured:

System / result	Cost of the LLM-heavy path	Source
Nemori vs SimpleMem (construction tokens)	`7.04M vs 1.3M (~5x)`	Anatomy of Agentic Memory
MemoryOS vs SimpleMem (end-to-end query latency)	`32.4 s vs 1.06 s`	Anatomy of Agentic Memory
A-MEM offline construction	`~15 h`	Anatomy of Agentic Memory
Memo memory-augmented query	`~7,000 tok/q at top-200`	Memo, State of AI Agent Memory 2026

Table 1. The write-time "LLM tax," from the published field. These are other systems' reported figures; they motivate a deterministic default and are not our measurements.

A fifth row has been removed from that table since this paper was first drafted. It reported a format-error rate for small backbones, attributed to Anatomy of Agentic Memory. A full-text search for the figure across roughly twenty-five candidate papers returns no hits and an exact-phrase web search returns nothing; it is not in the

paper it was attributed to. We withdraw it, and we have not substituted another figure, because we have not read a replacement in a primary source ourselves. It is recorded here rather than silently deleted because the failure was ours: we printed a number against someone else's published work without opening that work.

4 The read path: lookup-first with bounded escalation

A cheap deterministic retrieval path handles the common case. Tier 0 is always on and spends no model call: hybrid retrieval fusing vector, keyword, and graph signals. Structurally harder questions escalate in a fixed precedence, and each escalation tier costs exactly one model call, so the default path deliberately has none. Every tier is individually gated and degrades byte-identically to the tier below when disabled or when its model call fails. Figure 2 shows the ladder.

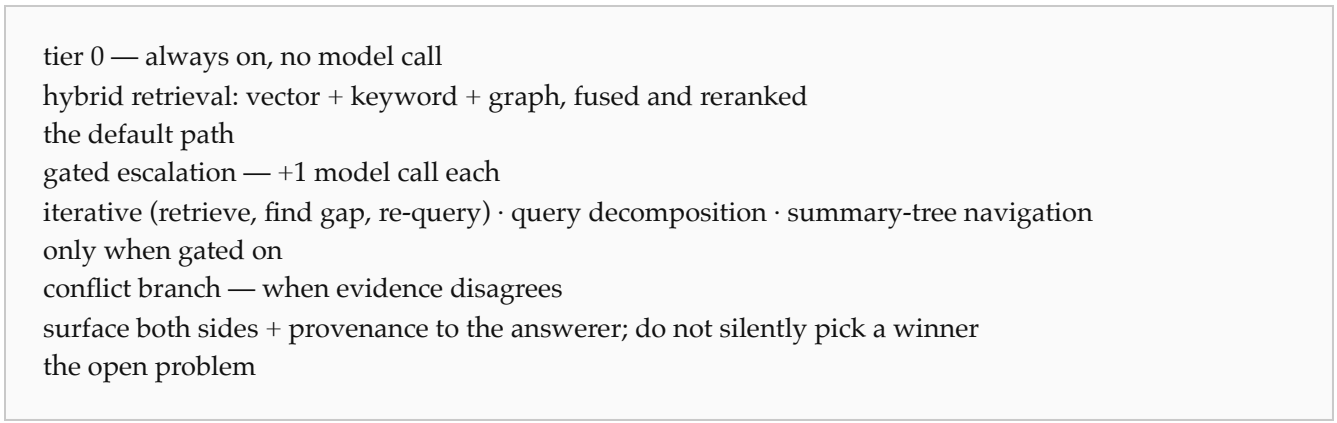

Figure 2. Lookup-first with bounded escalation. Tier 0 answers the common case with no model call; escalation tiers each cost one call and are individually gated. The conflict branch surfaces disagreement rather than hiding it.

The escalation shape is convergent with two independent papers on "lookup first, search as fallback": TierMem trades tokens for accuracy via a trained router, and Governed Memory caps reflection at two rounds and routes between a fast no-LLM hybrid path near 850 ms and a slower LLM path of 2 to 5 s. The iterative-retrieval tier is adopted from General Agentic Memory via Deep Research, whose plan-retrieve-reflect loop reports accuracy scaling with reflection depth. Tree navigation is a gated tier over our librarian summary tree rather than a replacement for retrieval.

The conflict branch is the piece the field has not solved. When retrieved evidence disagrees, we surface the conflict to the answerer with both sides and their provenance, and the model resolves it in context. This targets the competency the literature reports as universally failed: MemoryAgentBench makes conflict resolution one of its four competencies and finds that every method it tested fails the multi-hop case, reaching at most 6 percent accuracy, and contradiction resolution is the worst category in mem0's own published results at both scales. We report the mechanism here and hold our own conflict-resolution measurement to the publication gate until it clears a real sample size (section 10).

5 Consolidation off the hot path

Recursive summarization and reorganization run as a maintenance pass outside the query path, never during a read. The design intent is that a query reads a bounded, pre-consolidated view rather than re-deriving over the full history each time. The strongest published evidence that moving consolidation off the hot path is an order-of-magnitude cost win is LightMem, which beats a strong baseline on a long-memory benchmark while

cutting tokens 32 to 117 times and API calls 17 to 177 times. The same call was made independently at the largest deployed scale: OpenAI's June 2026 "Dreaming" moved ChatGPT memory to background consolidation for a self-reported roughly 5x compute reduction, the cut that made free-tier memory viable, and LangChain's hot-path-versus-background split and Letta's sleep-time reflection are the same idea arriving separately. The principled trigger, moving toward topic-boundary or semantic-divergence detection rather than token counts, is the direction our librarian is headed.

6 The head-to-head

We obtained mem0's open-source server and ran it beside Brain. Both engines went through the same mem0 benchmark harness, answered by the same `gpt-4o-mini` and judged by the same `gpt-4o-mini`, at the same `top_20` retrieval cutoff, on the same machine, with mem0-OSS served from its own container against Qdrant. Both sides' embedders were matched downward to a local MiniLM-384 model, so neither engine got a hosted-embedding advantage. The split is LoCoMo conversations 5 through 9, the held-out half; we tune on 0 through 4. That is 778 questions. No vendor-published number enters the comparison on either side.

Run against mem0's own open-source server on an identical harness, with the same answerer and judge, Brain scored 88.0% to mem0's 66.6% on 778 held-out LoCoMo questions, including 78.8% against 12.7% on temporal reasoning. Table 2 gives the breakdown by category.

Category	memo-OSS (run by us)	Brain
Single-hop	`82.0%`	`91.5%`
Multi-hop	`84.3%`	`90.7%`
Open-domain	`64.0%`	`82.0%`
Temporal	`12.7%`	`78.8%`
Overall, 778 questions	`66.6% (518/778)`	`88.0% (685/778)`

Table 2. Held-out LoCoMo conversations 5 to 9, 778 questions, both engines through the same harness with the same `gpt-4o-mini` answerer and judge at `top_20`, on one machine. memo-OSS is memo's open-source server run by us; it is not memo's published platform, which is a different system and reports a higher figure than either column here.

One category carries most of the margin, and saying so is part of reporting it. Temporal is 165 of the 778 questions, over a fifth of the benchmark, and mem0's open-source server answers 21 of them correctly. Temporal reasoning is the axis the last several development phases targeted deliberately, so a large gap there is the work landing rather than a surprise. In the other three categories the margin is considerably smaller.

One deviation must travel with the number. mem0's own harness container does not build: its requirements file pins their package to a git branch that no longer exists in their repository. We pinned `mem0ai==2.0.13` from PyPI instead. This is not the v3 pipeline they benchmarked, and a reader should treat that sentence as part of the figure rather than as a footnote to it. We are reporting what their published open-source server does when someone else installs it, which is a fair thing to measure and is not the same thing as their best configuration.

Two further caveats bound the result. LoCoMo's answer key is independently audited at 6.4 percent wrong, capping achievable accuracy near 93.6 percent; that ceiling does not threaten a margin of this size, but it does mean neither number is precise to the decimal and the decimals are decoration. And the Brain arm ran a pre-2026-07-25 configuration, because the run started before a reranker fix landed. The shipped configuration measures better on a controlled A/B, so this is a lower bound on the current system rather than a measurement of it.

What this is not: it is not that we beat mem0. It is mem0's open-source server, run by us, on one benchmark, on one split. mem0's published platform figure is higher than both columns in Table 2 and is a different system, and section 7 is the only place in this paper that compares anything against that platform.

A second run exists that puts both engines on one machine and adds wall-clock speed to the comparison. We are citing neither it nor any figure from it. Its run artifacts have not yet been committed to the research repository, and committing them is a stated condition on citing the result at all.

7 What a query costs, and a correction

Accuracy settled on one harness leaves the second question: what does a query cost. Against mem0's published configuration, 7,000 tokens per query at a top-200 retrieval budget, Brain answers from roughly 780 tokens of retrieved context at top-20, for about 2.4x lower cost per query. The clause naming which mem0 that is measured against is not a qualification of the ratio; it is part of it. mem0's published platform and mem0's open-source server are different systems with different retrieval budgets, and a ratio against one says nothing about the other.

A second cost comparison exists, against the open-source server we ran in section 6, and we are holding it internal. It is derived rather than measured on both sides: our own token counts come from provider-reported usage, but their in-container extraction cost is not instrumented, so one side of that ratio is an estimate built from a shared harness rather than a measurement. Publishing a cost ratio where only one side is measured is the mistake section 8 spends its length describing, and we are not going to make it in the same paper.

This section also corrects a claim of our own. This project previously quoted a token ratio against mem0 that came from a counter measuring retrieval payload only: no prompt scaffolding, no completions, and no call other than the answer. That is a real measurement of payload, it is not a cost ratio, and it was read as one, including by us. The counter has been relabelled as payload and a per-stage cost instrument built beside it, which is where the figure in the first paragraph comes from. The earlier ratio is withdrawn.

8 Benchmark doctrine: measuring in a credibility crisis

Section 1 set out why a leaderboard number is not evidence. The same problem applies to us, which is why our doctrine is organized around what we are allowed to claim rather than only around what we measure. Recall is also not use: MemoryArena embeds memory in interdependent multi-session tasks and reports that agents near-saturated on long-context memory benchmarks like LoCoMo perform poorly once the memory has to guide later actions. Architecture ideas from the literature are trustworthy; leaderboard deltas are not, including ours.

That doctrine is a three-tier suite in which conflating the tiers is exactly how a vendor ends up citing a saturated synthetic exam as a quality result. Figure 3 shows the split.

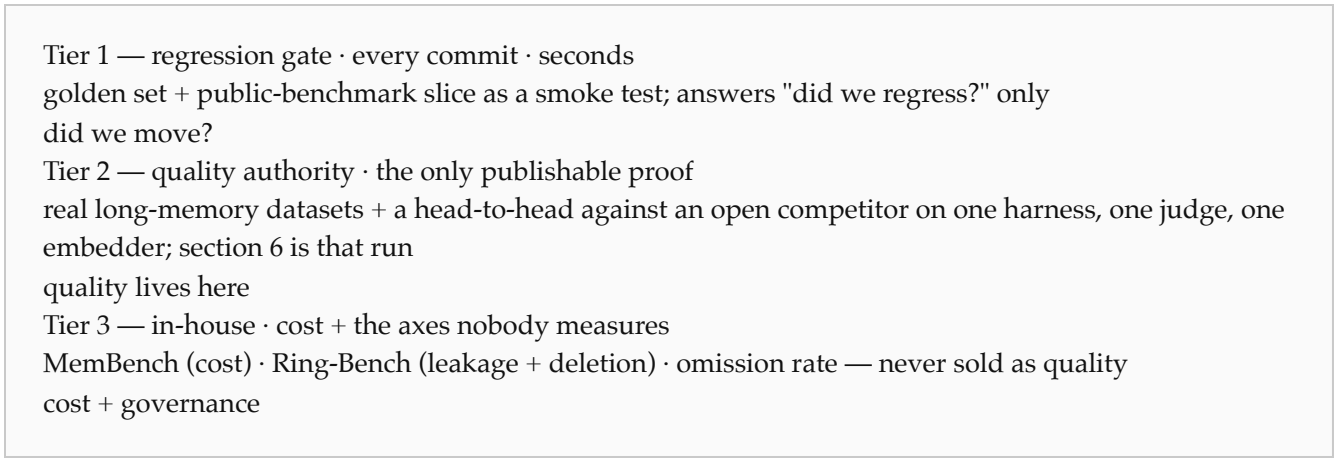

Figure 3. The three-tier benchmark suite. Tiers 1 and 2 prove competitiveness on the field's terms; Tier 3 owns the cost and governance axes the field has not learned to measure. Every Tier-3 run must pass a discrimination gate (score oracle, no-memory, and random-retrieval references alongside the real system) or declare itself not interpretable.

MemBench is the Tier-3 cost instrument, and its figures sit at internal in the claims ledger, so none of them are in this paper. The audit story is the better content anyway. A blind audit of MemBench caught a bug that would have tilted the comparison in our favour, an opt-in judge path that silently truncated the full-context baseline while still charging it full cost, found and fixed before the number left the building. A result that survives an adversary trying to bend it toward you is worth more than a flattering one that has not been attacked. Two earlier in-house benchmarks passed fairness audits and were still worthless as quality measures because they saturated, which is why the discrimination gate exists, why quality is confined to Tier 2, and why the result this paper leads with is a Tier-2 run against someone else's software rather than an in-house instrument.

9 The architecture was convergent

The four load-bearing choices, an append-only event log, bitemporal claims with supersession, deterministic-first writes, and provable deletion, were committed before the 2026 literature converged on all four, and section 2 of our research record documents that convergence paper by paper, including work that arrived after we shipped. The point of saying so is not credit; it is confidence. When independent groups arrive at the same substrate from event sourcing, from bitemporal databases, from the cost economics of the LLM tax, and from governance requirements, the substrate is more likely to be right than a leaderboard delta is. The governance layer, memory scoped per tenant with multi-agent rings governing what crosses between agents and deletion verified to a zero residual rather than assumed, is where the published field says enterprise value actually sits, and it is an axis almost nobody benchmarks; our Ring-Bench found real cross-tenant leaks in our own system on its first run, which is the entire reason to build the instrument.

10 What this version withdraws, and what it does not claim

An earlier version of this paper was named after a read-cost result: a growing-history benchmark on which Brain's per-query read cost was set beside a full-context baseline as history scaled. That result has been withdrawn to internal in our claims ledger. The measurement log outranks the ledger, and the hardening actions the log raises against that instrument are still open, so until they close the finding is directional rather

than citable. It does not appear here in any form, as a figure, as a ratio, or as a shape. The paper has been retitled rather than patched, because the withdrawn result was the old title and a paper cannot keep a thesis it has stopped being able to state.

Two further results that shaped the system are at internal for related reasons and are absent here rather than softened: a verdict on whether write-time LLM extraction earns its cost, which now needs re-measuring against a cost instrument that did not exist when the question was first answered, and an in-house cost-per-correct-answer benchmark. We name them instead of leaving holes, because a reader who saw the earlier version is owed the difference between what was removed and what was merely rewritten. Our doctrine is that a reversed conclusion stays visible with the thing that overturned it, and that applies to a paper as much as to a research record.

What is left is bounded, and the bounds are the interesting part. This paper claims one accuracy result, against one competitor's open-source server, on one harness, on one benchmark whose answer key is 6.4 percent wrong, with a dependency pin that is not the pipeline that competitor benchmarked. It claims one cost ratio, against that competitor's published platform configuration and against nothing else. It does not claim a speed result, a general cost result, a result about mem0 as a product, or that the margin would survive a different benchmark. Conflict resolution, reconciliation of changing facts, concurrent multi-writer memory, and the omission rate of our own write path are open axes we measure internally and do not publish until they are written up in full, including the parts that do not flatter us.

11 Conclusion

The useful thing about a benchmark result is not the number, it is whether a reader can tell what produced it. The field's published numbers do not survive being re-run, so we stopped citing them and ran the comparison ourselves: mem0's open-source server and Brain, one harness, one answerer, one judge, one machine, one held-out split. Brain scored 88.0% to mem0's 66.6% on 778 held-out LoCoMo questions, including 78.8% against 12.7% on temporal reasoning, with a dependency pin that is not the pipeline mem0 benchmarked and a Brain configuration that has since improved. Against mem0's published platform configuration a query reads from roughly 780 tokens rather than 7,000, for about 2.4x lower cost. Everything else this project has measured about its own cost currently sits at internal, and this version says so where the previous version said something stronger. The engine underneath is a diary and a fact sheet: an immutable log, claims that supersede rather than overwrite, and consolidation kept off the query path. On the evidence here, that shape is a claim about one comparison rather than about memory in general, and the next thing worth doing is the same exercise against a second system.

References

1. Naïve Research. *Brain: methodology, provenance, and benchmark doctrine; claims ledger*. Internal research record, usenaive/brain, July 2026.
2. mem0. *mem0 open-source memory server, mem0ai 2.0.13*. github.com/mem0ai/mem0 and PyPI, accessed July 2026.
3. Mem0. *State of AI Agent Memory 2026*. mem0.ai, April 2026.
4. Maharana, A. et al. *Evaluating very long-term conversational memory of LLM agents (LoCoMo)*. ACL 2024.
5. *The Log is the Agent: Event-Sourced Reactive Graphs for Auditable, Forkable Agentic Systems*. arXiv:2605.21997, 2026.
6. *ESAA: Event Sourcing for Autonomous Agents in LLM-Based Software Engineering*. arXiv:2602.23193, 2026.
7. Rasmussen, P. et al. *Zep: A Temporal Knowledge Graph Architecture for Agent Memory*. arXiv:2501.13956, 2025.
8. Yadav, D. *Revisiting Zep's 84% LoCoMo claim: corrected evaluation and 58.44% accuracy*. GitHub issue, getzep/zep-papers #5, 2025 — together with Zep's subsequent correction of its own figure to 75.14%.
9. *Memory in the LLM Era: Modular Architectures and Strategies in a Unified Framework*. arXiv:2604.01707, 2026.

10. *Anatomy of Agentic Memory: Taxonomy and Empirical Analysis of Evaluation and System Limitations*. arXiv:2602.19320, 2026.
11. *Convomem Benchmark: Why Your First 150 Conversations Don't Need RAG*. arXiv:2511.10523, 2025.
12. *From Lossy to Verified: A Provenance-Aware Tiered Memory for Agents (TierMem)*. arXiv:2602.17913, 2026.
13. *General Agentic Memory via Deep Research*. arXiv:2511.18423, 2025.
14. *Governed Memory: A Production Architecture for Multi-Agent Workflows*. arXiv:2603.17787, 2026.
15. *Evaluating Memory in LLM Agents via Incremental Multi-Turn Interactions (MemoryAgentBench)*. arXiv:2507.05257, 2025.
16. *MemoryArena: Benchmarking Agent Memory in Interdependent Multi-Session Agentic Tasks*. arXiv:2602.16313, 2026.
17. *LightMem: Lightweight and Efficient Memory-Augmented Generation*. arXiv:2510.18866, 2025.
18. Snodgrass, R. T. *Developing Time-Oriented Database Applications in SQL*. Morgan Kaufmann, 2000.

The head-to-head in section 6 was run by us on both sides. mem0's open-source server was pinned to `mem0ai==2.0.13` from PyPI because mem0's own harness container does not build, so it is not the v3 pipeline they benchmarked. Both arms used the same `gpt-4o-mini` answerer and judge at `top_20` on one machine, over LoCoMo conversations 5 to 9, with embedders matched downward to a local MiniLM-384 model. LoCoMo's answer key is independently audited at 6.4 percent wrong. The Brain arm ran a pre-2026-07-25 configuration, so its figure is a lower bound. The cost figure in section 7 is against mem0's published platform configuration and is not a ratio against their open-source server, which is a different system. Field and competitor figures elsewhere are the vendors' or authors' own published numbers, cited for context and not re-run by us. Correspondence: team@usenaive.ai.