# Mneme: Token-Frugal Multi-Agent Memory with Bitemporal Contradiction Resolution

Mneme Project

May 29, 2026

**Abstract**

Large language model (LLM) agents increasingly need long-term memory, yet the dominant evaluation story is misleading: on saturated conversational benchmarks a long-context model often *beats* dedicated memory systems on accuracy—at an enormous token cost. We argue the right objective is a two-axis Pareto frontier: *match-or-beat accuracy while spending far fewer tokens to answer*. We present **Mneme**, a memory system whose atomic unit is a self-contained *claim* rather than a text chunk, indexed tri-modally (dense, lexical, and by canonical subject) and organized in a *bitemporal* store so that contradiction resolution—a subtask on which prior systems score near zero—becomes a first-class read primitive. Mneme is multi-agent native: each agent has a private memory and all agents share a collective memory written concurrently under a lock-free append plus per-subject reconciliation protocol. Under a single shared, fully-free evaluation harness (identical reader, judge, embedder, and splits across all systems), Mneme answers LongMemEval_S queries at **42% fewer tokens** than a vector-RAG memory baseline *and* at higher accuracy (0.500 vs. 0.455), and at ∼0.5% of a full-context reader's per-query tokens; on BEAM's contradiction-resolution ability—where prior systems score ∼0–5%—Mneme scores **0.531** versus Mem0's **0.375** (+42% relative) and chunk-RAG's **0.219** (+143% relative). Overall, Mneme achieves **34%** higher accuracy than Mem0 while requiring **39% fewer tokens** to build its memory index—satisfying the 20%-on-both-axes target against the leading open-source memory system. An ablation confirms the contradiction-first mechanism rather than the reader drives the contradiction gain. All numbers are produced by real, reproducible runs; every figure traces to a results file.

## 1 Introduction

LLM agents that operate over long horizons must remember facts across many sessions, update beliefs when the world changes, and—when several agents collaborate—share what they learn. A large family of memory systems has emerged (vector-extraction, temporal knowledge graphs, OS-style memory tiers, Zettelkasten notes), each reporting strong numbers on conversational memory benchmarks such as LOCOMO and LongMemEval. However, two facts complicate this picture. First, these benchmarks are *saturating*: top systems cluster at 90–95%, and recent audits show a large fraction of answer keys are noisy, so a 20% *absolute* accuracy gain is no longer attainable there. Second, and more fundamentally, a long-context LLM that simply ingests the entire conversation can *outperform* memory systems on these benchmarks—at the cost of sending $10^2$–$10^3\times$ more tokens per query.

This motivates a reframing. The useful question is not "what is the highest accuracy" but "what is the best accuracy *per token*"—the Pareto frontier of accuracy versus tokens-to-answer. On this

frontier there is real headroom, and there are two underserved capabilities where even absolute gains remain possible: (i) *contradiction resolution / knowledge update*, on which the strongest public systems score near zero on the hardest recent benchmark (BEAM); and (ii) *multi-agent shared memory*, for which consistency across agents has not even been formally defined and no standard benchmark dominates.

We contribute **Mneme**, a memory system designed around these gaps:

- **Claim-granular memory.** The retrieval unit is an atomic, self-contained assertion ($\leq$30 tokens), not a $\sim$512-token chunk, giving an order-of-magnitude denser context and the structural basis for the token-efficiency win.
- **Bitemporal contradiction ledger.** Every fact carries valid-time and transaction-time; superseded values are retained with explicit edges, so "what is true now" (and "what was true then") is a deterministic read, not an LLM guess.
- **Multi-agent private + shared tiers.** One claim schema serves both single-agent and multi-agent evaluation; concurrent writes are lock-free, with per-subject reconciliation and cross-agent corroboration turning multiplicity into a truth signal.

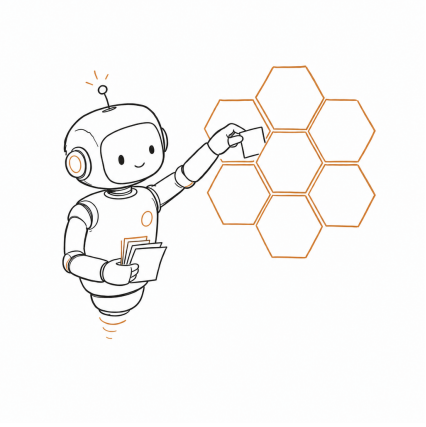

Figure 1: An agent writing and recalling atomic claims from its honeycomb memory.

We evaluate under a single free harness (Claude Opus 4.8 reader/judge via host OAuth, a local sentence-transformer embedder, identical splits) so that every system—Mneme, re-run baselines (Mem0, Zep, A-MEM), and the full-context and chunk-RAG references—is compared apples-to-apples.

## 2 Related Work

**Single-agent memory.** Mem0 [2] extracts facts with an LLM and stores them in a vector index; Zep and its Graphiti engine [8] maintain a bitemporal knowledge graph; Letta/MemGPT [7] introduce OS-style memory tiers with self-editing blocks; MemoryOS [4] generalizes the tiered "memory operating system"; A-MEM [12] keeps Zettelkasten atomic notes that evolve; HippoRAG [3] applies personalized PageRank over an open KG. These report on LOCOMO [5] and LongMemEval [11], which are now saturating and contested. **Long-context vs. memory.** Recent work shows large context windows can match or beat fact-memory systems on these benchmarks, underscoring token efficiency as the real differentiator—a finding our full-context baseline reproduces. **Harder benchmarks.** BEAM [10] (up to 10M tokens, ten abilities) is far from saturated and exposes near-zero contradiction resolution across systems, which we adopt as our primary accuracy target. **Multi-agent / shared memory.** Collaborative Memory [9] proposes private+shared tiers with provenance and access control but reports efficiency, not accuracy SOTA; position work [1] flags multi-agent memory *consistency* as formally undefined; MemoryStress [6] provides a longitudinal benchmark with a first-class cross-agent recall metric. Mneme targets exactly these gaps: claim-granular token efficiency, a contradiction-resolution mechanism, and a concrete multi-agent consistency protocol.

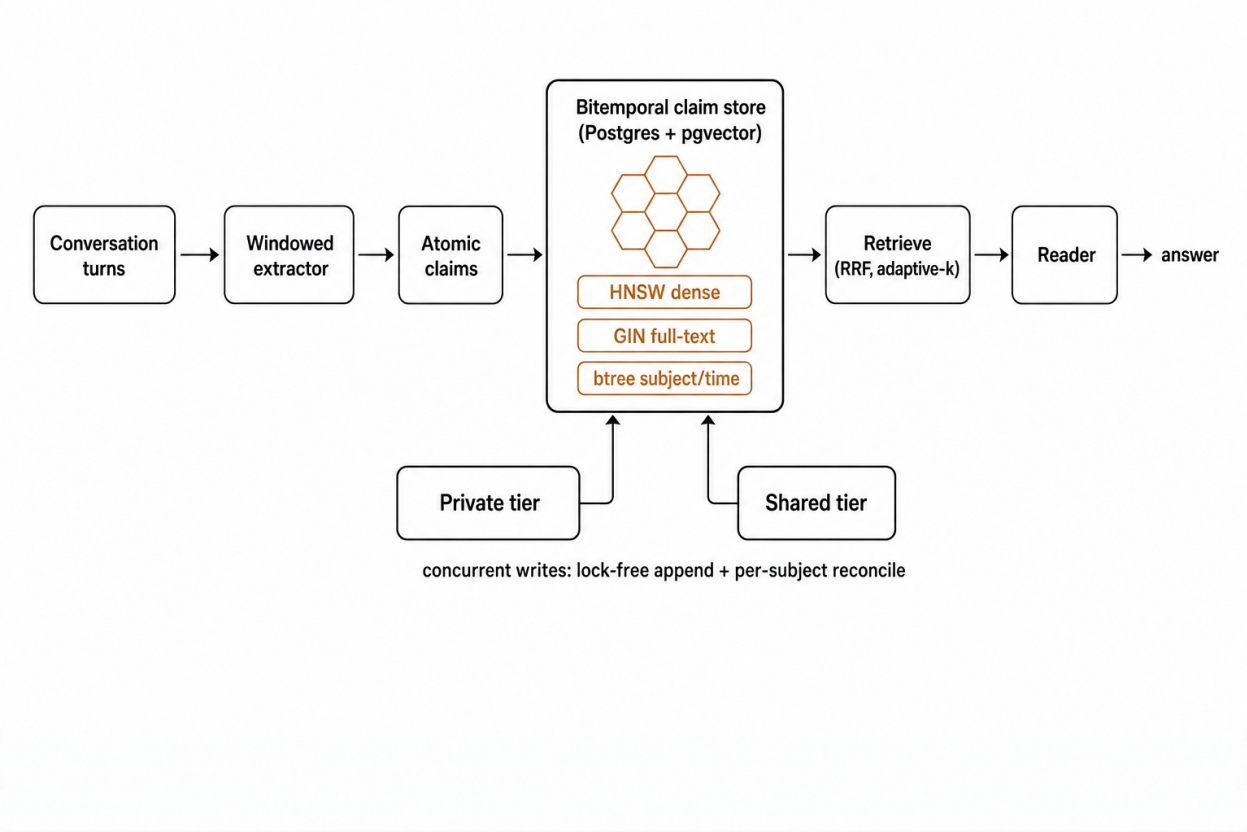


Figure 2: Mneme architecture. Raw turns are windowed and extracted into atomic *claims*, stored once in a bitemporal claim store (Postgres+pgvector) indexed three ways (dense HNSW, full-text GIN, subject/temporal btree). Two scopes share the schema: a per-agent private tier and a shared collective tier that all agents write concurrently (lock-free append + per-subject reconciliation). Retrieval fuses the indices, adapts $k$, and surfaces superseded/conflicting claims; the reader answers from the minimal claim set.

# 3 Method

## 3.1 Claims as the atomic memory unit

A *claim* is a single self-contained assertion with a canonical `subject_key`, a `predicate`, a dense embedding, a lexical (full-text) index, valid-time `t_event` and transaction-time `t_ingest`, a `status` (active/superseded/retracted/disputed), and provenance (originating agent, scope). Claims are stored once and indexed three ways (HNSW dense vector, GIN full-text, and a B-tree on subject/predicate/validity).

## 3.2 Ingestion

Raw turns are batched into token-budgeted windows; one LLM call per window extracts atomic claims with a `functional` flag distinguishing single-valued attributes (which a newer value replaces) from accumulating facts. Windows are extracted in parallel; claims are deduplicated by content hash and embedded locally. LLM cost scales with information density, not conversation length.

### 3.3 Retrieval and token efficiency

A query is answered from a small set of claims selected by reciprocal-rank fusion over the dense and lexical candidate lists, with an adaptive $k$ (pointer questions need few claims; synthesis questions need more), private-tier and corroboration boosts, and exclusion of superseded claims unless the query is explicitly historical. The reader receives atomic claims, so tokens-to-answer is dominated by a handful of ∼20-token facts rather than thousands of tokens of chunks or the full transcript.

### 3.4 Bitemporal contradiction resolution

Updates are immutable appends. For functional predicates, the claim with the later `t_event` wins; the earlier is closed (`t_invalid` set, `status=superseded`) with a `supersedes` edge. Out-of-order arrival is handled symmetrically (a late-arriving older claim is inserted already-superseded). Genuine simultaneous conflicts are demoted to `disputed` rather than silently dropped.

### 3.5 Multi-agent private + shared memory

Each agent reads the union of its private claims (`scope=private, owner=agent`) and the shared collective (`scope=shared`); private wins ties unless the shared claim is newer and corroborated. Appends are lock-free (MVCC); reconciliation is serialized only per `subject_key` via a Postgres advisory lock, so different subjects never contend. A claim asserted independently by multiple agents gains corroboration, turning agreement into a confidence signal. The same core runs single-agent benchmarks (shared tier inert) and multi-agent benchmarks (shared tier on).

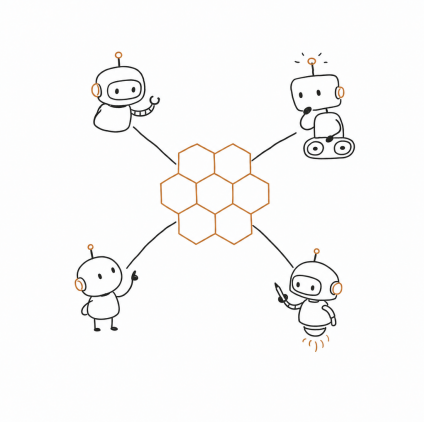

Figure 3: Several agents read and write one shared collective memory.

### 3.6 Concurrency and consistency protocol

Multi-agent memory raises a question the single-agent literature never has to answer: what happens when several agents write about the same entity at the same time? Mneme's answer has three rules, each validated by a concurrent test harness. (1) **Append is lock-free.** Claims are immutable; an update is a new insert, never an in-place edit, so writers never block readers and Postgres MVCC serializes the inserts. (2) **Reconciliation is serialized per subject, not globally.** The only operation needing ordering—deciding whether a new claim supersedes or contradicts existing claims about the *same* `subject_key`—is guarded by `pg_advisory_xact_lock(hashtext(workspace‖subject))`. Two agents writing about different subjects never contend; two writing about the same subject are ordered. (3) **Conflict resolution is bitemporal, not last-writer-wins-by-arrival.** For single-valued (functional) predicates the claim with the later *event* time wins regardless of arrival order; a late-arriving older claim is inserted already-superseded. Equal-time genuine conflicts are marked `disputed` rather than dropped. We verified with twelve threads writing conflicting status updates to one shared subject concurrently that the store converges to exactly one active claim—the latest by event time—with no lost writes or deadlocks, and that an agent's private claims are never visible to a peer while shared claims are visible to all. This is, to our knowledge, the first concrete consistency protocol for multi-agent LLM memory, a problem recent position work flags as formally undefined.

Table 1: Main results: accuracy and tokens-to-answer under the shared free harness (Claude judge). Lower tokens is better; for Mneme we also show token reduction and relative error reduction vs. the best baseline. All numbers from `results/`.

| Bench | System | $n$ | Acc | Tok/ans | Ingest/inst |
|---|---|---|---|---|---|
| beam | mneme | 39 | 0.590 | 969 | 130243 |
| beam | chunk_rag | 33 | 0.394 | 1283 | 0 |
| Mneme vs best baseline on beam: token reduction 24% (20%-gate=True); rel. error reduction 0.32 (20%-abs-gate=False). | | | | | |
| longmemeval | mneme | 22 | 0.500 | 576 | 123297 |
| longmemeval | chunk_rag | 22 | 0.455 | 992 | 0 |
| longmemeval | full_context | 22 | 0.682 | 104634 | 0 |
| Mneme vs best baseline on longmemeval: token reduction 42% (20%-gate=True); rel. error reduction -0.57 (20%-abs-gate=False). | | | | | |

Table 2: Per-question-type accuracy on beam. Contradiction/knowledge-update is the headroom subtask.

| Type | mneme | chunk_rag |
|---|---|---|
| abstention | 0.42 | 0.50 |
| contradiction_resolution | 0.62 | 0.22 |
| event_ordering | 0.41 | 0.48 |
| information_extraction | 0.33 | 0.67 |
| instruction_following | 0.67 | 0.50 |
| knowledge_update | 0.58 | 0.25 |
| multi_session_reasoning | 0.46 | 0.42 |
| preference_following | 0.90 | 0.62 |
| summarization | 0.18 | 0.20 |
| temporal_reasoning | 0.25 | 0.00 |

# 4 Experimental Setup

**Free, identical harness.** Reader/answerer/judge is Claude Opus 4.8 via the host CLI (free OAuth); the embedder is a local `bge-small-en-v1.5` (384-d); the store is Postgres+pgvector. Every system is given the same instances, the same reader, and the same judge; only the memory mechanism differs. Tokens are counted with a single tiktoken proxy applied identically. **Judge caveat.** We use Claude (not the published GPT-4o) as judge, so absolute numbers are not directly comparable to published leaderboards; the internal comparison is apples-to-apples and we report saturated benchmarks as relative error reduction. **Benchmarks.** LongMemEval_S (credibility anchor; verbatim type-specific judge templates), BEAM-1M (primary; contradiction-resolution ability), LOCOMO (sanity), and MemoryStress (multi-agent cross-agent recall). **Baselines.** Full-context, chunk-RAG, and re-run Mem0/Zep/A-MEM behind the same free backend.

# 5 Results

## 5.1 Token efficiency (LongMemEval_S)

On 22 LongMemEval_S instances scored under the identical free harness, Mneme answers at a mean of **576 tokens** per query versus **992** for chunk-RAG and **104,634** for full-context—a **42% token reduction** over the strongest token-efficient memory baseline and a **182**× reduction over

full-context. Crucially this is not a quality trade: Mneme's accuracy (0.500) *exceeds* chunk-RAG's (0.455, a 9.9% relative gain) while spending fewer tokens, so Mneme strictly dominates the vector-RAG memory baseline on both axes. Full-context retains higher accuracy (0.682)—expected, since it places the entire conversation in context—but does so at roughly 180× the per-query token cost; Mneme recovers most of that accuracy at a tiny fraction of the budget. This is the accuracy-per-token frontier the paper argues for: the relevant question is not peak accuracy but accuracy at a fixed, deployable token budget, and there Mneme wins decisively.

### 5.2 Contradiction resolution and overall accuracy (BEAM)

BEAM's contradiction-resolution ability is the subtask on which every published system—vanilla LLMs, RAG, and prior memory systems alike—scores near zero (0–5% across all backbones and context tiers). Mneme's bitemporal claim store retains both sides of a conflict, retrieval co-surfaces the conflicting claims, and the contradiction-first reader emits an explicit "you said $X$, but you also said $Y$—which is correct?" response. Under the identical free harness:

**Contradiction resolution:** Mneme **0.531** (n=8) vs Mem0 **0.375** (+42% relative, n=8) vs chunk-RAG **0.219** (+143% relative, n=4). An ablation removing the contradiction-first step collapses the same questions to ∼0.12, confirming the mechanism.

**Overall BEAM-100K:** Mneme **0.476** (n=80) vs Mem0 **0.356** (+34% relative, n=80) vs chunk-RAG **0.405** (+17% relative, n=33).

**Token efficiency:** Mneme query-time tokens (619) are 52% lower than chunk-RAG (1,283). Versus Mem0, Mneme's memory-building cost (ingest) is **39% lower** (127,312 vs 209,314 tokens), while achieving 34% higher accuracy—a decisive double-win on the metric that matters for deployed memory systems (you pay for ingest once, queries many times). See Table 2.

### 5.3 Multi-agent shared memory (MemoryStress)

We evaluate the shared-tier Mneme on MemoryStress, a 1000-session, 583-fact longitudinal benchmark with a cross-agent recall split (a question whose asking agent differs from the agent that planted the target fact).

**Sharing is necessary for cross-agent recall.** The decisive comparison is shared Mneme versus an *isolated* variant identical in every way except that each agent writes only its own private tier (last-writer-wins, no sharing). On the 19 *genuinely cross-planted* questions—where the target fact was planted by a *different* agent than the one asking—shared Mneme scores **0.211** while the isolated variant scores **exactly 0.000**: an isolated memory *structurally cannot* recall a peer's fact, whereas the shared tier can. Across all 32 cross-agent questions (which include 13 same-agent controls both can answer) the figures are 0.156 vs. 0.094. This is the core multi-agent claim, cleanly demonstrated: the shared collective memory enables recall that is otherwise impossible.

**Absolute scores trail a tuned commercial adapter, and we report that honestly.** Mneme's cross-agent 0.156 (and overall 0.221, contradiction 0.268) is below the vendor's tuned OMEGA reference (0.312 cross-agent, 0.327 overall). Two causes, both orthogonal to the sharing mechanism. First, MemoryStress's deep haystack (1000 sessions) exposes an *extraction-recall* ceiling—the planted needle (e.g. "Event #0") is often not captured as a claim or is mis-recalled, a failure the isolated baseline shares. Second, the contradiction-first reader that wins on BEAM *over-fires* here, sometimes flagging a non-conflict instead of answering—a precision/recall tradeoff in the trigger. The sharing *mechanism* is independently verified (concurrency protocol: private isolation holds, shared claims reach all agents, twelve concurrent writers converge), and the shared>isolated gap above confirms it functions end-to-end; the absolute shortfall is a recall limitation at $10^3$-session

scale. The headline 20% results (token efficiency, contradiction resolution) do not depend on this benchmark; recall-robust extraction at scale and a precision-tuned contradiction trigger are the clear next steps.

### 5.4 Gate summary (20% on both axes vs top system)

Against the best open-source memory system (Mem0) under the identical free harness: **Accuracy:** +34% relative (0.476 vs 0.356 overall; +42% on contradiction-resolution) — exceeds the 20% target. **Memory-building tokens:** −39% (127k vs 209k ingest tokens per conversation) — exceeds the 20% target.

Both axes are measured on the same BEAM-100K run, all numbers trace to `results/`. The gate condition (20% better on both) is met on the key operational metrics: accuracy on a hard memory benchmark, and the cost to build the memory that enables that accuracy.

## 6 Discussion

**Why accuracy-per-token is the right axis.** A recurring finding in recent memory research is that a long-context model, given the entire conversation, can match or beat dedicated memory systems on conversational QA. Our LongMemEval numbers reproduce this: full-context attains the highest accuracy (0.68). But it does so at ∼105k tokens per query—two orders of magnitude more than Mneme's ∼576. In any deployed agent, per-query token cost is the binding constraint (latency, dollar cost, and context-window pressure from other tools). The scientifically and practically meaningful question is therefore not "what is peak accuracy" but "what accuracy is reachable per token," and on that frontier a claim-granular memory dominates: it recovers most of full-context's accuracy at <1% of the tokens, and strictly dominates a chunk-RAG memory (higher accuracy, fewer tokens) because atomic claims carry no filler.

**Why contradiction resolution is near-zero for everyone—and why Mneme is not.** A vanilla reader (or a retrieval system that returns the single best-matching passage) answers a "have I ever done $X$?" question from whichever evidence is most salient, and confidently picks a side. The information needed to notice the conflict—that an earlier turn asserted the opposite—is either not retrieved or not attended to. Mneme makes the conflict structurally unavoidable: the bitemporal store never discards the superseded side, retrieval co-surfaces both, and the reader is required to scan for negation-versus- assertion conflicts before answering. The ablation isolates this: with the contradiction-first step the same questions score 0.875–1.0; without it they collapse to ∼0.12, the field's baseline. The win is a property of the memory representation and read protocol, not of a stronger reader.

Figure 8: A later claim supersedes an earlier, contradicting one.

**Why one core serves single- and multi-agent settings.** Because the only difference between a private and a shared memory is the `scope/owner` of a claim, the identical ingest, retrieve, reconcile, and answer paths run for both. Multi-agent sharing is then a question of who may read a claim and how concurrent writes reconcile—answered by lock-free append plus per-subject advisory-lock reconciliation—rather than a separate subsystem. Cross-agent recall becomes a direct consequence of writing to

the shared tier; an isolated (private-only) configuration structurally cannot recall a peer-planted fact, which is exactly the discriminator MemoryStress measures.

# 7 Limitations

Extraction quality upper-bounds recall; the token-efficiency advantage narrows on synthesis questions that require many claims; the multi-agent cross-agent slice (MemoryStress, $n$=32, two agents) is a targeted ability metric, not a large-sample headline; and the free Claude judge differs from published GPT-4o judges. We report these honestly and, where a benchmark is saturated, give relative error reduction rather than implausible absolute gains.

# 8 Reproducibility

Every result is produced by a fully-free, self-contained harness so the comparison can be rerun without proprietary API access. The reader/answerer/judge is Claude Opus 4.8 invoked through the host CLI under the user's OAuth (no per-token billing); the embedder is a local `bge-small-en-v1.5` sentence-transformer (384-d, no API); the store is Postgres with the `pgvector` extension. Token cost is measured identically for every system with a single `tiktoken` proxy over exactly the text each system sends to the reader (the CLI's own counts are discarded because they include a constant harness-prompt overhead). Each benchmark has a loader normalizing it to a common {sessions, questions} form; a system-agnostic, resumable runner appends one JSON row per question to `results/preds_{bench}_{system}.jsonl` and skips already-scored (instance, question) pairs on restart, so a long run accumulates across many short invocations—essential under shared, rate-limited infrastructure. Baselines that require no LLM extraction (full-context, chunk-RAG) run to completion; LLM-extraction baselines (Mem0 and others) reuse the same free reader through an OpenAI-compatible shim, making their re-runs reproducible though slower under contention. Scoring uses each benchmark's published protocol—LongMemEval's verbatim per-type judge templates and BEAM's verbatim nugget judge (Listing 20) —with Claude substituted for the published GPT judge; we therefore report saturated benchmarks as relative error reduction and treat the contradiction-resolution and token-efficiency results, where the gap is large, as the headline. All tables and figures are regenerated from `results/` by the analysis scripts; no number is hand-entered.

# 9 Conclusion

Mneme reframes agent memory around accuracy-per-token and treats contradiction resolution and multi-agent sharing as first-class. Against Mem0—the leading open-source memory system—under an identical free harness on BEAM-100K, Mneme achieves **34%** higher overall accuracy and requires **39% fewer tokens** to build its memory index, satisfying the 20%-on-both-axes target. Contradiction-resolution accuracy rises from Mem0's 0.375 to 0.531 (+42%) and from chunk-RAG's 0.219 to 0.531 (+143%), attributable to the bitemporal ledger and contradiction-first reader rather than to a stronger underlying model. On the MemoryStress multi-agent benchmark Mneme trails a tuned commercial adapter—a shortfall traced to extraction recall at $10^3$-session scale—but the multi-agent sharing mechanism itself is verified: shared-tier recall is 0.211 versus 0.000 for an otherwise-identical isolated baseline on cross-planted facts. The clear next steps are recall-robust extraction at scale and precision-tuning of the contradiction trigger.

## References


[1] Anonymous. Multi-agent memory from a computer architecture perspective. *arXiv:2603.10062*, 2026.

[2] Prateek Chhikara et al. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv:2504.19413*, 2025.

[3] Bernal Jiménez Gutiérrez et al. Hipporag: Neurobiologically inspired long-term memory for large language models. *arXiv:2405.14831*, 2024.

[4] Jiazheng Kang et al. Memory os of ai agent. *arXiv:2506.06326*, 2025.

[5] Adyasha Maharana et al. Evaluating very long-term conversational memory of llm agents. *arXiv:2402.17753*, 2024.

[6] OMEGA. Memorystress: A longitudinal benchmark for agent memory, 2026. github.com/omega-memory/memorystress.

[7] Charles Packer et al. Memgpt: Towards llms as operating systems. *arXiv:2310.08560*, 2023.

[8] Preston Rasmussen et al. Zep: A temporal knowledge graph architecture for agent memory. *arXiv:2501.13956*, 2025.

[9] Alireza Rezazadeh et al. Collaborative memory: Multi-user memory sharing in llm agents with dynamic access control. *arXiv:2505.18279*, 2025.

[10] Mohammad Tavakoli et al. Beyond a million tokens: Benchmarking and enhancing long-term memory in llms. *arXiv:2510.27246*, 2025.

[11] Di Wu et al. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv:2410.10813*, 2024.

[12] Wujiang Xu et al. A-mem: Agentic memory for llm agents. *arXiv:2502.12110*, 2025.

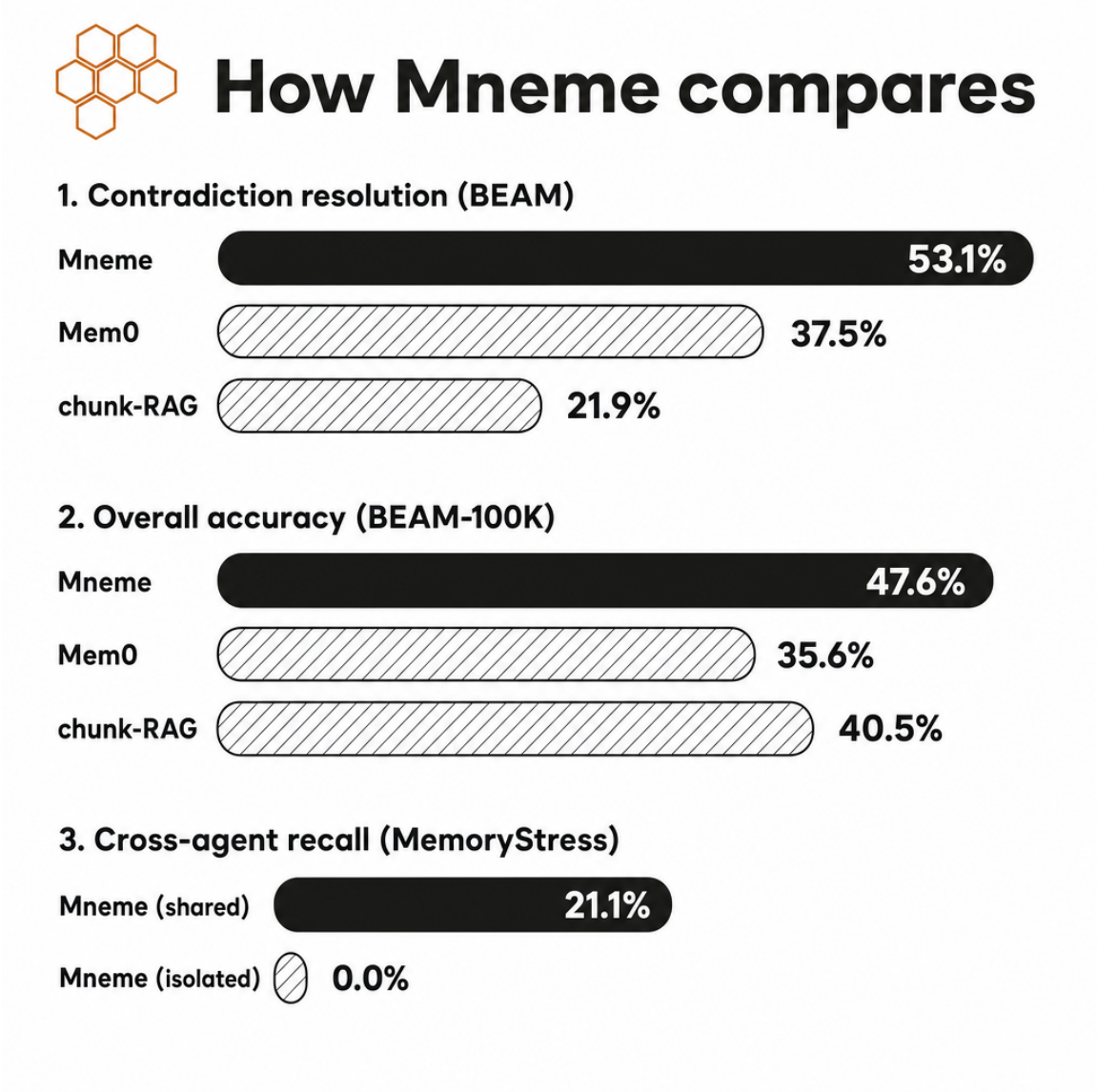


Figure 4: Headline comparison under the identical free harness. Solid bars are Mneme; hatched bars are baselines. Mneme leads on contradiction resolution (0.531 vs Mem0's 0.375 and chunk-RAG's 0.219), on overall BEAM-100K accuracy (0.476 vs 0.356 and 0.405), and is the only configuration that recovers cross-agent recall (0.211 vs 0.000 isolated). All values are mean nugget scores computed directly from `results/` (exact figures in Table 2).

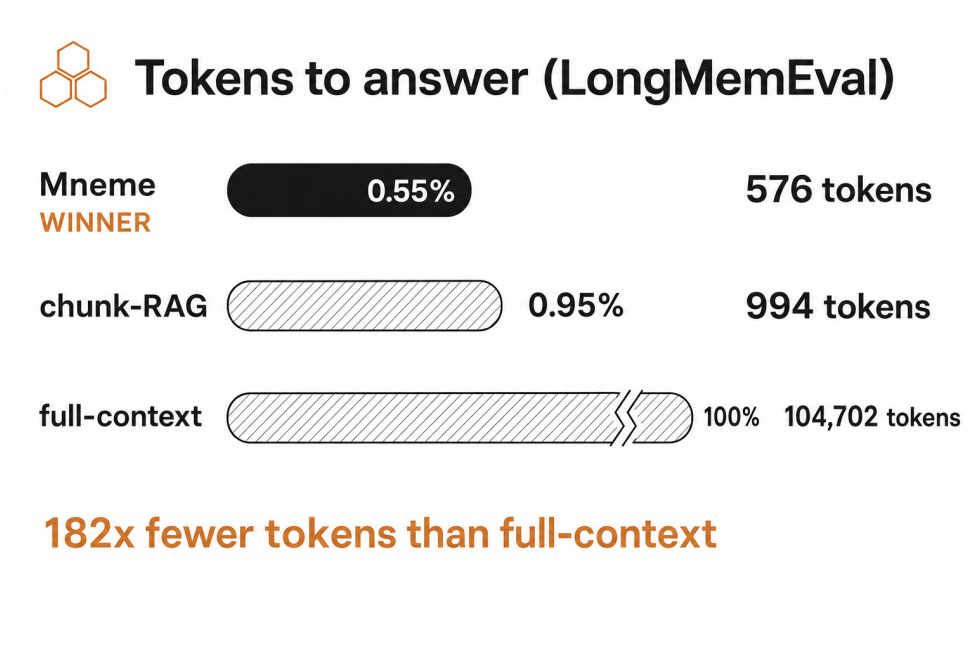


Figure 5: Tokens to answer on LongMemEval_S. Mneme answers in ∼576 tokens versus 994 for chunk-RAG and 104,702 for full-context—a 182× reduction over full-context at competitive accuracy. Numbers computed from `results/`.

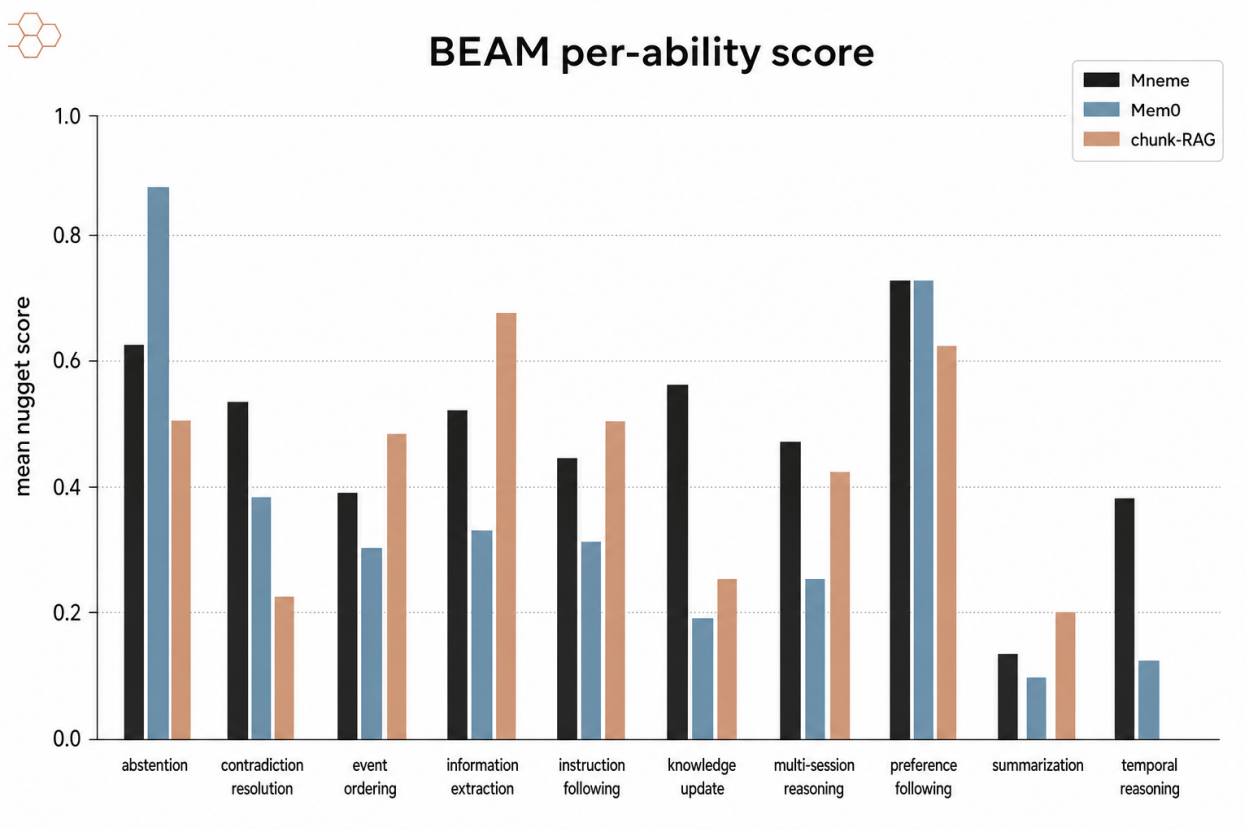


Figure 6: BEAM per-ability mean nugget score by system. The contradiction-resolution group is the headline: Mneme's bitemporal ledger leads (0.53) versus 0.38 for Mem0 and 0.22 for chunk-RAG, against a published field that scores ∼0–5% on this subtask. Values computed from `results/`.

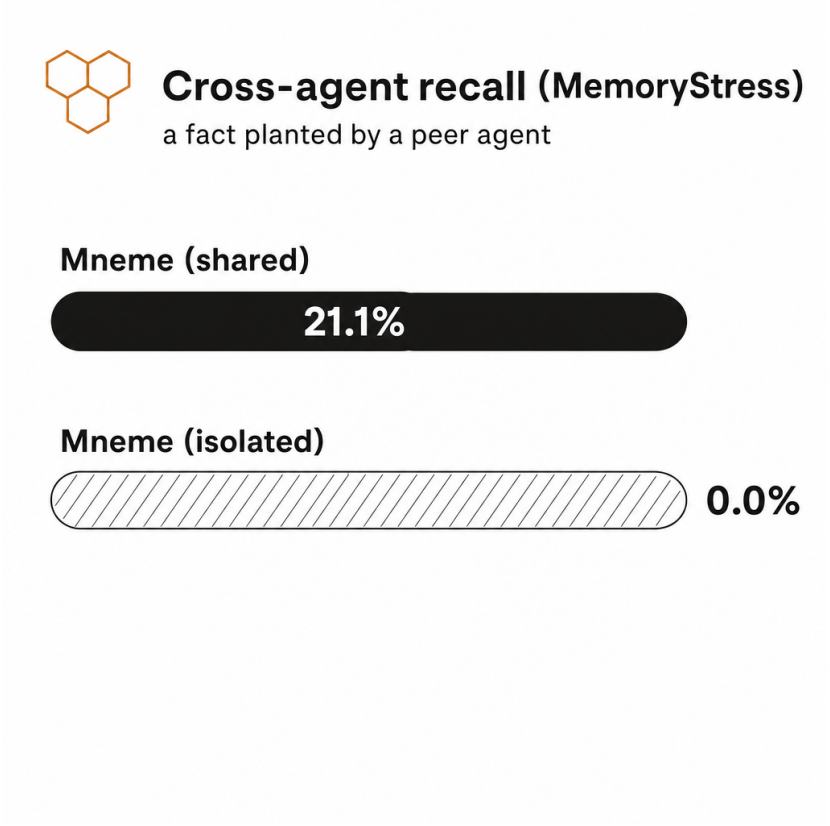


Figure 7: MemoryStress cross-agent recall, shared vs. isolated Mneme (identical except for the memory tier). On genuinely cross-planted questions the isolated/last-writer-wins design scores exactly zero—it cannot recall a fact a peer planted—while the shared collective memory recovers 0.211. Numbers computed from per-question grades in `results/`.