MEMSTATE: DYNAMIC MEMORY STATE GRAPH CONSTRUCTION AND AGENTIC STATE GRAPH RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory systems must answer questions about matters whose states evolve across extended interactions. Memory systems that overwrite or delete superseded values lose access to earlier states. Questions about historical trajectories, reasons for change, and final outcomes can require both intermediate states and the current state. We propose MEMSTATE, a long-term memory management framework that preserves state evolution in a dynamic graph and retrieves it through an agentic process. Dynamic Memory State Graph Construction (DMSG) preserves successive states and distinguishes direct state transitions from related contextual information. Chain-level summaries guide incremental linking and reduce semantic drift across long state chains. Agentic State Graph Retrieval (ASGR) treats initial retrieval as a partial observation, organizes memories into matter-level units, and uses frontier-based staleness signals with multi-round tool calls to recover historical paths, subsequent changes, and the latest state. MEMSTATE achieves 76.0% complete evidence recall and 85.0% history recall, improving over A-MEM by 11.0 and 13.0 percentage points, respectively. Evaluations on RHELM, LoCoMo, and LongMemEval-S show gains in temporal reasoning, multi-hop question answering, and knowledge updates. Construction and retrieval ablations support the roles of state preservation, chain-level linking, and adaptive graph traversal in evidence recovery.

1 INTRODUCTION

Long-term memory enables language-model agents to maintain continuity across conversations, tasks, and changing user circumstances. Retrieval-augmented models make external information accessible beyond a model's parameters, while memory systems select and organize information accumulated during interaction (Lewis et al., 2020; Park et al., 2023; Packer et al., 2023). Their usefulness depends on recalling what happened to a particular matter over time. A maintenance request, a travel plan, or a personal preference can pass through several states before reaching its present condition. Answering questions about these matters requires a memory representation that retains their evolution.

State evolution makes long-term memory a management problem as well as a retrieval problem. An updated value can correctly describe the present while omitting the evidence needed to explain an earlier decision or reconstruct a sequence of changes. Conversely, a relevant retrieved memory can describe a state that has already been superseded. Figure 1 illustrates these two failure modes. Our key question is how a memory system can preserve and recover the evolution of the same matter, including its historical states and recorded outcome, when only a limited amount of evidence can enter the answer context.

Existing approaches address complementary aspects of this problem. Vector and hybrid retrieval rank individual records, hierarchical methods summarize large memory collections, and update-based systems maintain compact representations through extraction, consolidation, and revision (Karpukhin et al., 2020; Sarthi et al., 2024; Zhong et al., 2024; Chhikara et al., 2025). Graph-based memories introduce relational structure, including associative links and temporal relations (Gutiérrez et al., 2024; Xu et al., 2025; Rasmussen et al., 2025). These advances motivate a more

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

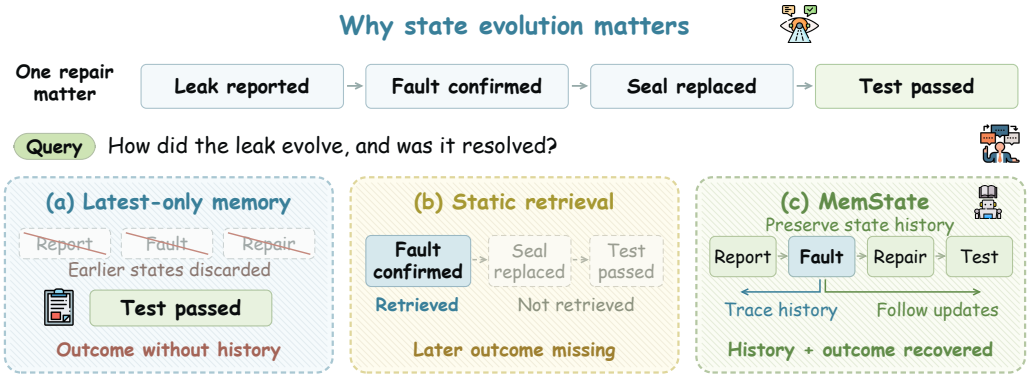


Figure 1: Why state evolution matters. In this illustrative repair history, latest-only memory loses earlier states, while a partial static retrieval misses subsequent updates. MEMSTATE preserves the state graph and traces historical and later evidence to recover the trajectory and outcome. Icons adapted from Huang et al. (2025).

explicit treatment of matter-level evolution. A memory system must distinguish an actual state transition from background information, preserve historical evidence through updates, and identify which parts of a retrieved matter remain outside the current observation.

Two challenges connect state construction to state retrieval. During construction, locally plausible links can gradually join different matters, particularly when long chains share vocabulary and entities. During retrieval, a small set of relevant records provides incomplete visibility into the matter's graph. It may exclude earlier explanations, subsequent developments, or an entire branch. We address these challenges through consistent matter assignment during construction and evidence expansion during retrieval. Preserving states alone does not ensure that retrieval will recover the needed historical evidence; expanding a graph with incorrect matter assignments can introduce unrelated evidence.

We propose MEMSTATE, which combines Dynamic Memory State Graph Construction (DMSG) with Agentic State Graph Retrieval (ASGR). DMSG retains state nodes, separates strong transition edges from weak contextual edges, and uses chain-level summaries to constrain incremental graph construction. ASGR groups retrieved nodes into matter-level units, identifies units whose retrieved frontier has unseen successors, and exposes graph traversal through multi-round `DAG_Search` calls. The agent traces history, advances toward the latest state, and follows branches according to intermediate observations. This design connects preserved historical evidence with subsequent changes and outcomes through adaptive retrieval.

Our contributions are fourfold. **Dynamic state management** provides a unified representation of historical states, state progression, and related context. **DMSG construction** combines relation strength with chain-level semantic anchoring to reduce drift during incremental linking. **ASGR retrieval** couples matter units and frontier-based staleness with multi-round graph tools to recover missing state evidence. **State-centered evaluation** examines answer quality, evidence completeness, graph construction, robustness, and cost across three long-term memory benchmarks.

2 RELATED WORK

We review external retrieval, persistent agent memory, structured memory, and tool-mediated reasoning, focusing on how these approaches represent changing information and retrieve it for subsequent queries.

External retrieval connects language models to information beyond their parameters. REALM, RAG, and dense passage retrieval develop retrieval-based access to textual knowledge, while RETRO integrates large-scale retrieval into language modeling (Guu et al., 2020; Lewis et al., 2020; Karpukhin et al., 2020; Borgeaud et al., 2022). Large context windows offer another route to historical information, but evidence placement affects utilization and retrieval remains relevant to inference cost (Liu

et al., 2024; Li et al., 2024). MEMSTATE organizes evidence selection around the evolution of an identified matter.

Persistent agent memory develops mechanisms for storage, consolidation, reflection, and updating. Generative Agents organize observations and reflections, MemGPT manages movement between memory tiers, and MemoryBank models persistent interaction memory (Park et al., 2023; Packer et al., 2023; Zhong et al., 2024). Mem0 supports scalable extraction and updates, while MemOS and MIRIX explore broader memory management architectures (Chhikara et al., 2025; Li et al., 2025; Wang & Chen, 2025). LightMem and SimpleMem emphasize efficient memory processing and compression; Memory-R1 and MEM1 study learning memory decisions or maintaining memory for extended reasoning (Fang et al., 2026; Liu et al., 2026; Yan et al., 2025; Zhou et al., 2025). Our focus is the coexistence and retrieval of successive states of the same matter.

Structured memories add organization beyond independent text records. RAPTOR builds recursive summaries, GraphRAG supports graph-based summarization, and HippoRAG develops associative retrieval with graph representations (Sarthi et al., 2024; Edge et al., 2024; Gutiérrez et al., 2024; Jiménez Gutiérrez et al., 2025). A-MEM links and evolves memory items, Zep explicitly represents temporal knowledge, and Hindsight studies structured memory for agent reasoning (Xu et al., 2025; Rasmussen et al., 2025; Latimer et al., 2025). In MEMSTATE, the graph distinguishes state advancement from same-matter context, and the retrieval process uses the boundary of the observed graph to guide further exploration.

Tool-mediated reasoning supplies a mechanism for adapting retrieval to intermediate observations. ReAct interleaves reasoning and actions, and Toolformer studies language-model use of external tools (Yao et al., 2023; Schick et al., 2023). Long-term memory benchmarks such as LoCoMo, LongMemEval, and RHELM examine cross-session evidence, temporal reasoning, updates, and heterogeneous histories (Maharana et al., 2024; Wu et al., 2025; Zhang et al., 2026). MEMSTATE uses an explicit state graph and an agentic traversal interface. Its evaluation measures both the evidence recovered from that graph and the answers produced from the recovered context.

3 METHOD

MemState manages long-term memory as an evolving collection of related states and retrieves these states through successive graph observations. Dynamic Memory State Graph Construction (DMSG) preserves memory increments, distinguishes state transitions from contextual associations, and maintains a summary for each matter. Agentic State Graph Retrieval (ASGR) groups initial retrieval results into state units, identifies units with unseen successors, and invokes graph search across multiple rounds.

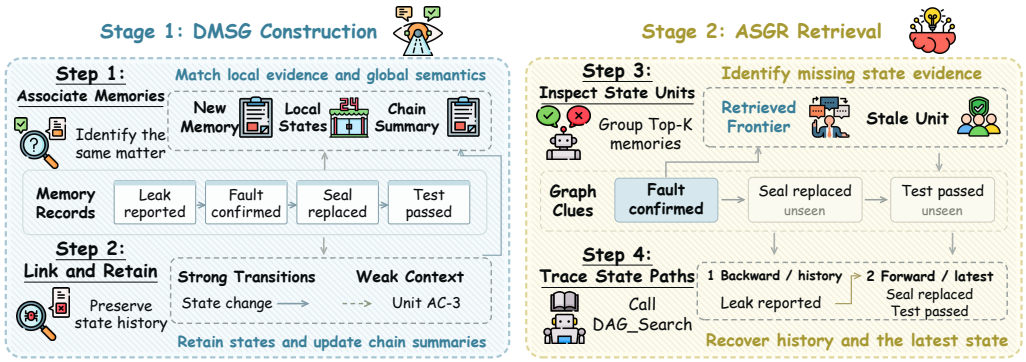
3.1 OVERVIEW AND NOTATION

Figure 2 separates memory construction from query-time exploration. In the construction stage, DMSG incorporates each incoming memory and updates the summary of its associated matter. In the retrieval stage, ASGR starts from conventional similarity retrieval and expands its partial observations through `DAG_Search`. The resulting evidence is passed to a reader under a context budget. Algorithm 1 presents the two stages and their shared graph representation.

Let x_t denote the t-th incoming memory increment, and let v_t denote its state node. The index t records ingestion order; temporal information about the described events remains associated with the memory. After t increments, the representation is

$$G_t = (V_t, E_t^{\mathrm{s}}, E_t^{\mathrm{w}}, \mathcal{S}_t), \qquad V_t = \{v_1, \ldots, v_t\}, \tag{1}$$

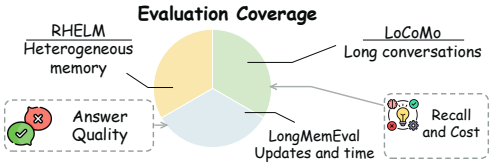
where E_t^{s} and E_t^{w} contain strong and weak relations, respectively, and $\mathcal{S}_t$ contains the chain summary nodes. A connected component $C \subseteq V_t$ represents a matter. Connectivity groups the state nodes, while the directed successor structure supports traversal. We write $s_{C,t}$ for the summary associated with the corresponding matter chain. At query time, G_T denotes the graph containing the available memory.



(a) DMSG preserves state evolution; ASGR recovers missing evidence through successive graph observations.



(b) Evidence for state-aware answers.



(c) Benchmark coverage, shown schematically.

Figure 2: MEMSTATE overview. (a) DMSG preserves an illustrative maintenance history with typed links and updated summaries; ASGR detects unseen successors and retrieves history and later states through successive `DAG_Search` calls. (b) Evidence supports state-aware answers. (c) Three benchmark settings. Visual layout and icon cutouts adapted from Figure 2 of Huang et al. (2025).

3.2 DYNAMIC MEMORY STATE GRAPH CONSTRUCTION

Preserving states and typed relations. DMSG retains an incoming state alongside the states that preceded it. A strong relation indicates that one memory directly advances the same matter, such as moving from identifying a fault to confirming its cause and then deploying a repair. A weak relation connects memory about that matter without asserting direct advancement, including background details, parallel activities, and repeated records. Strong transition edges follow the directed state order and are constrained to remain acyclic; weak contextual edges do not participate in successor traversal. The graph supports DAG search over state progression and retains contextual links for evidence assembly.

The insertion operation preserves both the state records and the relations already constructed. Writing ΔE_t^j for the newly introduced relations of type j, the state graph evolves as

$$V_t = V_{t-1} \cup \{v_t\}, \qquad E_t^j = E_{t-1}^j \cup \Delta E_t^j, \quad j \in \{\mathrm{s}, \mathrm{w}\}. \tag{2}$$

Historical states remain individually retrievable after later updates.

Chain summaries for state assignment. The chain summary describes a matter across its stored states. DMSG evaluates an incoming memory against individual nodes and the chain summary when deciding where it belongs. The assignment thus incorporates context beyond adjacent states.

We denote this joint attachment decision and the subsequent relation assignment by

$$\begin{aligned} A_t &= \mathrm{Associate}(x_t, V_{t-1}, \mathcal{S}_{t-1}), \\ (\Delta E_t^{\mathrm{s}}, \Delta E_t^{\mathrm{w}}) &= \mathrm{Relate}(x_t, A_t). \end{aligned} \tag{3}$$

Here A_t denotes the selected matter and relevant existing states; an incoming memory can also initiate a matter. Relation assignment distinguishes direct advancement from same-matter context.

For an incoming memory assigned to C, DMSG updates the matter summary as

$$s_{C,t} = \mathrm{UpdateSummary}(s_{C,t-1}, x_t). \tag{4}$$

The first memory initializes a chain's summary. Updating this summary changes the chain's global description while Equation (2) preserves its state records.

3.3 AGENTIC STATE GRAPH RETRIEVAL

State units from local observations. ASGR interprets conventional retrieval results as partial observations of the constructed graph. Given query q, an initial retriever returns the Top-K state nodes $R^{(0)}$. Let $\mathcal{C}$ be the connected components of the state graph. The retrieved portion of each observed matter and the resulting collection of state units are

$$R_C^{(0)} = R^{(0)} \cap C, \qquad \mathcal{U}^{(0)} = \{(C, R_C^{(0)}) : C \in \mathcal{C},\ R_C^{(0)} \neq \varnothing\}. \tag{5}$$

Retrieved nodes from the same matter form one unit, which the agent can revisit across calls. At round r, $R^{(r)}$ contains the nodes retrieved so far, and $R_C^{(r)} = R^{(r)} \cap C$ records the visible portion of C.

Retrieved frontiers and stale units. A retrieved frontier identifies the most advanced states currently visible within a matter. Let $v \prec_C u$ denote the reachability ordering induced by directed strong transition edges in component C; weak contextual edges connect related records but do not define successor order. The frontier consists of the maximal retrieved nodes under that ordering,

$$F_C^{(r)} = \{v \in R_C^{(r)} : \nexists u \in R_C^{(r)} \text{ with } v \prec_C u\}. \tag{6}$$

For a linear chain, this set reduces to its latest retrieved state. For a branching DAG, several retrieved states can remain maximal simultaneously.

A unit is marked stale when one of its retrieved frontier nodes has an unseen successor in the full component. Formally,

$$\text{stale}_C^{(r)} = \mathbf{1}\left[\exists v \in F_C^{(r)},\ \exists u \in C \setminus R_C^{(r)} \text{ such that } v \prec_C u\right]. \tag{7}$$

The stale indicator applies to the whole state unit and is provided with its summary and retrieved frontier.

Multi-round graph exploration. The agent invokes `DAG_Search` to explore state units. Each call specifies a unit, a direction selected from forward, backward, and both, and a target selected from latest, history, full graph, and branch. Optional starting nodes identify where traversal begins; a hop limit and a per-call budget control the requested exploration. The tool returns nodes, edges, a graph frontier, the unseen-successor count, a stale indicator, and the matter summary.

Each observation informs the next retrieval decision. Let $H^{(r)}$ be the previous calls and their returned observations, and let $a^{(r)}$ be the next tool invocation selected by the agent. An executed round has the form

$$\begin{aligned} a^{(r)} &= \text{Agent}(q, \mathcal{U}^{(r)}, H^{(r)}), \\ o^{(r)} &= \texttt{DAG_Search}(a^{(r)}; G_T), \\ R^{(r+1)} &= R^{(r)} \cup \text{Nodes}(o^{(r)}). \end{aligned} \tag{8}$$

The returned evidence updates the state units, frontiers, stale indicators, and call history. The agent can follow later changes, recover preceding states, or inspect another branch according to what the previous call revealed. Exploration ends when the agent obtains sufficient evidence, selects a stopping action, or reaches the maximum of L calls.

In Figure 2, the retrieved frontier is a confirmed fault, while the repair and test result remain unseen. ASGR first retrieves the preceding report, then follows the later states to recover the repair and successful test.

The reader receives the evidence assembled after $r_\star$ executed calls, where $0 \leq r_\star \leq L$. With final-evidence budget B, the answering interface is

$$\mathcal{E}_q = \text{Assemble}(q, R^{(r_\star)}), \quad \text{Tokens}(\mathcal{E}_q) \leq B, \quad \hat{y} = \text{Reader}(q, \mathcal{E}_q). \tag{9}$$

Algorithm 1 MemState construction and retrieval

Require: Memory increments $x_{1:T}$, query q, retrieval size K, final-evidence budget B, round limit L

Initialize G_0 and its chain summaries
Stage 1. Dynamic memory state graph construction
for $t = 1, \ldots, T$ **do**
  Associate x_t with a matter and assign typed relations — Eq. (3)
  Retain v_t and its new relations — Eq. (2)
  Initialize or update the associated chain summary — Eq. (4)
end for
Stage 2. Agentic state graph retrieval
$R \leftarrow \mathrm{TopK}(q, G_T, K)$; $H \leftarrow \varnothing$; $r \leftarrow 0$
while $r < L$ and the agent continues retrieval **do**
  Group R into units; compute frontiers and stale marks — Eqs. (5)–(7)
  Select the next `DAG_Search` call or stop using q, units, and H
  if the selected action is stop **then**
    break
  end if
  Execute the call and add returned nodes to R — Eq. (8)
  Append the call and its observation to H; $r \leftarrow r + 1$
end while
Assemble evidence within B and return the reader's answer — Eq. (9)

4 EXPERIMENTS

We evaluate benchmark QA, state-evidence retrieval, graph construction, robustness, efficiency, and deletion. Appendix B provides the full results.

4.1 EVALUATION SETTING

We evaluate MEMSTATE on RHELM, LoCoMo, and LongMemEval-S using their native QA metrics (Zhang et al., 2026; Maharana et al., 2024; Wu et al., 2025). RHELM uses accuracy over heterogeneous memory sources, LoCoMo uses token-level F1, and LongMemEval-S uses LLM-judged accuracy. State-focused QA uses a shared binary rubric, macro-averaged first over query types and then over datasets.

Construction, retrieval decisions, and reading use Qwen3.5-27B in non-thinking mode with temperature zero and BF16. Embeddings and reranking use Qwen3-Embedding-8B and Qwen3-Reranker-8B; GPT-5.6 provides LLM judgments. Seeds are 13, 37, and 71. Initial retrieval returns 30 candidates and up to ten seed nodes. Final evidence is capped at 8,000 tokens and cumulative tool-return text at 12,000 tokens. ASGR permits up to four calls with at most ten hops per call. Appendix A specifies checkpoints and accounting.

ER@B and CER@B measure supporting content in the final evidence context. Source-level ER@10 and CER@10 deduplicate and truncate the final source ranking separately. Latest-state and history recall use their respective nonempty gold-evidence slices. Trajectory metrics evaluate transitions, ordered paths, branches, temporal order, and final-outcome support.

4.2 BENCHMARK AND STATE RETRIEVAL RESULTS

MEMSTATE achieves 35.0 overall and 52.0 temporal accuracy on RHELM. LoCoMo single-hop, multi-hop, temporal, and open-domain F1 are 47.0, 35.0, 57.0, and 49.0. LongMemEval-S overall accuracy is 88.0, with 92.0 knowledge-update and 91.0 temporal accuracy. Relative to the strongest controlled comparators, the temporal gain is 3.0 points on RHELM; LoCoMo multi-hop and temporal F1 improve by 1.0 each, and LongMemEval-S update and temporal accuracy improve by 2.0 each.

Table 1: Benchmark results under the unified protocol. The baseline is the highest-scoring controlled comparator for each metric. All scores are percentages; full comparisons are in Appendix B.

Benchmark	Metric	Baseline	Baseline score	MEMSTATE
RHELM	Overall accuracy	Zep/Graphiti	34	35
	Temporal accuracy	Zep/Graphiti	49	52
LoCoMo	Single-hop F1	Zep/Graphiti	47	47
	Multi-hop F1	Zep/Graphiti	34	35
	Temporal F1	Zep/Graphiti	56	57
	Open-domain F1	Zep/Graphiti	50	49
LongMemEval-S	Overall accuracy	Zep/Graphiti	88	88
	Knowledge update	Zep/Graphiti	90	92
	Temporal accuracy	Zep/Graphiti	89	91

Table 2: State evidence and answer quality. ER and CER use the final evidence context ($@B$). Latest and History measure their respective state recall. All scores are percentages; Calls counts graph-tool calls per question.

Method	ER	CER	Latest	History	QA	Calls
Hybrid RAG	70	49	68	61	70	0
RAPTOR	74	55	74	66	74	0
Mem0	77	60	86	49	75	0
A-MEM	80	65	83	72	79	0
MEMSTATE	88	76	93	85	86	2.2

MEMSTATE reaches 88.0 ER, 76.0 CER, 93.0 latest-state recall, and 85.0 history recall, exceeding A-MEM by 8.0, 11.0, 10.0, and 13.0 points. State-focused QA reaches 86.0, a 7.0-point gain. QA for current states, historical states, reasons for change, complete evolution, and final outcomes is 94.0, 88.0, 83.0, 75.0, and 90.0. Transition recall, path EM, branch recall, temporal-order accuracy, and outcome support reach 83.0, 67.0, 74.0, 81.0, and 91.0. Path EM and branch recall each improve by 23.0 points over A-MEM.

4.3 CONSTRUCTION AND RETRIEVAL ABLATIONS

With the strong-edge graph fixed, adding weak context raises history recall from 77.0 to 85.0 and QA from 82.0 to 86.0. Typing relations raises QA from 81.0 to 86.0. Chain summaries raise component purity from 82.0 to 91.0. At 32 hops, full DMSG assigns 83.0% of states correctly, compared with 66.0 for summary matching and 53.0 for local linking. On lexically similar but distinct matters, DMSG attains a 7.0% false-merge rate and 88.0% true-link recall. Removing summary-consistency rejection reduces strong-edge precision from 93.0 to 88.0, component purity from 91.0 to 85.0, and QA from 86.0 to 81.0.

Iterative retrieval recovers further evidence within fixed budget caps. Round limits of zero, one, two, and four achieve 70.0, 80.0, 84.0, and 86.0 QA, with CER of 49.0, 65.0, 73.0, and 76.0. The four-round limit uses 2.2 graph calls on average. Frontier staleness detection achieves 88.0% precision, recall, and F1, with an 8.0% false-positive rate.

The preservation controls vary access to historical states. Strict latest-only memory removes superseded states and accessible copies, yielding zero historical-evidence recall with both static retrieval and ASGR. A separate graph-structure-by-traversal test preserves all states: both static conditions reach 70.0 QA, local chronological chains with ASGR reach 81.0, and DMSG with ASGR reaches 86.0. CER increases from 69.0 to 76.0 when DMSG replaces local chains.

A compute-matched text-search control and ASGR each average 21,000 online model tokens per question. ASGR improves QA from 82.0 to 86.0 and CER from 70.0 to 76.0. ASGR also exceeds fixed-direction graph expansion by 4.0 QA points (Appendix B).

Table 3: DMSG ablations. Strong P and Weak P are relation-type precision; Purity measures state-component purity. Scores are percentages. Relation-type precision is inapplicable to untyped relations.

Construction variant	Strong P	Weak P	Purity	History	QA
Full DMSG	93	88	91	85	86
Without weak context	93	–	91	77	82
Untyped relations	–	–	84	80	81
Without chain summary	86	81	82	79	80
Without consistency rejection	88	82	85	81	81
Summary matching only	90	85	86	79	81
Local-node matching only	84	80	79	74	77

Table 4: ASGR round limits under the same evidence-budget caps. ER, CER, recall, and QA are percentages. Calls is the average number of graph-tool calls. The evidence and cumulative tool-return caps are 8K and 12K tokens.

Maximum rounds	ER	CER	Latest	History	QA	Calls
0	70	49	68	61	70	0
1	82	65	85	75	80	1
2	86	73	91	82	84	1.7
4	88	76	93	85	86	2.2
6	89	77	93	86	86	2.6
8	89	77	93	86	85.5	2.9

4.4 Generalization and Robustness

The QA gap over A-MEM increases across update-density groups. MemState and A-MEM both reach 84.0 QA without updates; with at least eight updates, their scores are 80.0 and 61.0. At 100K records, latest-state recall is 89.0 for MemState, 74.0 for Mem0, and 58.0 for Hybrid RAG. With retrieval evidence fixed, gains over Mem0 are 13.0, 11.0, and 6.0 QA points for Qwen3.5-9B, Qwen3.5-27B, and GPT-5.6 readers. Graph expansion improves ER by 16.0–22.0 points across retrieval backends.

From 86.0 unperturbed QA, removing 10% of strong edges yields 81.0 and adding 10% erroneous strong edges yields 76.0. Seven-day timestamp noise after construction yields 88.0 latest-state recall and 81.0 QA; removing all retrieval-time timestamps yields 78.0 and 74.0 (Appendix B.5).

4.5 Efficiency and Deletion

Limits of six and eight rounds yield 86.0 and 85.5 QA with 2.6 and 2.9 graph calls. The default averages 2.2 graph calls, 3.2 agent calls, and 4.2 online LLM calls. DMSG construction uses three LLM calls and 3,600 input-plus-output tokens per new memory.

Online inference averages 21,000 model tokens per question: 7,600 evidence tokens, 500 other reader-input tokens, 300 reader-output tokens, 12,000 agent-input tokens, and 600 agent-output tokens. Tool returns average 5,200 tokens per question and are included in model input when read. End-to-end latency is 36 seconds at p50 and 80 seconds at p95; offline judging is accounted for separately.

Cascading deletion covers source records, graph relations, summaries, derived indexes, and caches. Full cascading deletion achieves a 99.5% deletion pass rate, with 0.5% request-level residue and 0.1% query-level residue. Human evidence-support annotations have 85.0% agreement and Cohen's $\kappa = 0.70$. Detailed cost, deletion, and annotation results appear in Appendix B.

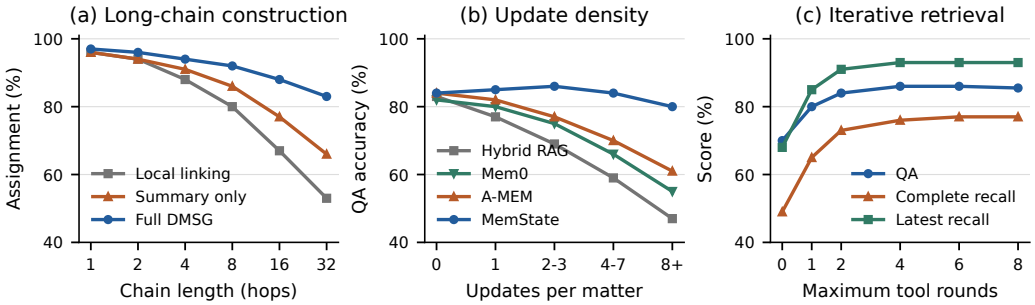


Figure 3: State complexity and traversal depth. Left, chain assignment as hop count increases. Middle, state QA by update density. Right, QA, complete evidence recall, and latest-state recall under increasing call limits.

5 Conclusion

We introduced MemState to preserve the evolution of matters in long-term memory and recover their history and outcomes through agentic graph retrieval. DMSG separates transitions from context and uses chain summaries to stabilize incremental construction. ASGR traverses the graph using retrieved frontiers and intermediate observations. MemState reaches 76.0% complete evidence recall and 85.0% history recall, improving over A-MEM by 11.0 and 13.0 points. Construction, factorial, and compute-matched evaluations connect these gains to state preservation, accurate linking, and adaptive retrieval. The QA gain over A-MEM increases across the tested update-density groups, and a four-call limit achieves the highest observed QA score.

AI Use Statement

AI assistance was used for translation, manuscript drafting and editing, method formalization, organization of results, literature curation, and preparation of source code and figures.

Ethics Statement

Long-term agent memory may contain personal information. Deployments should establish authorization, retention, and deletion policies, and answers should be grounded in traceable supporting evidence. Access to and redistribution of research benchmark data follow their original terms. State-transition relationships should be distinguished from causal explanations of real-world events.

Reproducibility Statement

Section 3 specifies the state graph, incremental construction, retrieval frontier, and iterative traversal. Appendix A provides the models, budgets, evidence units, and metric definitions; Appendix B provides detailed experimental results; and Appendix C provides published benchmark references and evaluation details.

References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2206–2240. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/borgeaud22a.html.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory, 2025. URL https://arxiv.org/abs/2504.19413.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv preprint arXiv:2404.16130*, 2024. URL https://arxiv.org/abs/2404.16130.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and Efficient Memory-Augmented Generation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.18866.

Bernal J. Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 59532–59569, 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3929–3938. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/guu20a.html.

Chao Huang, Benfeng Wang, Jie Wen, Chengliang Liu, Wei Wang, Li Shen, and Xiaochun Cao. Vad-R1: Towards Video Anomaly Reasoning via Perception-to-Cognition Chain-of-Thought, 2025. URL https://arxiv.org/abs/2505.19877.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2502.14802.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Chris Latimer, Nicoló Boschi, Andrew Neeser, Chris Bartholomew, Gaurav Srivastava, Xuan Wang, and Naren Ramakrishnan. Hindsight is 20/20: Building Agent Memory that Retains, Recalls, and Reflects, 2025. URL https://arxiv.org/abs/2512.12818.

Patrick Lewis, Ethan Perez, Aleksandra 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 *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Zhiyu Li, Chenyang Xi, Chunyu Li, Ding Chen, Boyu Chen, Shichao Song, Simin Niu, Hanyu Wang, Jiawei Yang, Chen Tang, Qingchen Yu, Jihao Zhao, Yezhaohui Wang, Peng Liu, Zehao Lin, Pengyuan Wang, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhen Tao, Huayi Lai, Hao Wu, Bo Tang, Zhengren Wang, Zhaoxin Fan, Ningyu Zhang, Linfeng Zhang, Junchi Yan, Mingchuan Yang, Tong Xu, Wei Xu, Huajun Chen, Haofen Wang, Hongkang Yang, Wentao Zhang, Zhi-Qin John Xu, Siheng Chen, and Feiyu Xiong. MemOS: A Memory OS for AI System, 2025. URL https://arxiv.org/abs/2507.03724.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. Retrieval augmented generation or long-context LLMs? a comprehensive study and hybrid approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 881–893, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.66. URL https://aclanthology.org/2024.emnlp-industry.66/.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. SimpleMem: Efficient Lifelong Memory for LLM Agents, 2026. URL https://arxiv.org/abs/2601.02553.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

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*, 2023. URL https://arxiv.org/abs/2310.08560.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A Temporal Knowledge Graph Architecture for Agent Memory, 2025. URL https://arxiv.org/abs/2501.13956.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2401.18059.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Yu Wang and Xi Chen. MIRIX: Multi-Agent Memory System for LLM-Based Agents, 2025. URL https://arxiv.org/abs/2507.07957.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.10813.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic Memory for LLM Agents. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2502.12110.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. Memory-R1: Enhancing Large Language Model Agents to Manage and Utilize Memories via Reinforcement Learning, 2025. URL https://arxiv.org/abs/2508.19828.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Han Zhang, Zihao Tang, Xin Yu, Xiao Liu, Yeyun Gong, Haizhen Huang, Yan Lu, Weiwei Deng, Feng Sun, Qi Zhang, and Hanfang Yang. Beyond Static Dialogues: Benchmarking Realistic, Heterogeneous, and Evolving Long-Term Memory, 2026. URL https://arxiv.org/abs/2605.31086. Accepted to Findings of EMNLP 2026.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing Large Language Models with Long-Term Memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946. URL https://doi.org/10.1609/aaai.v38i17.29946.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. MEM1: Learning to Synergize Memory and Reasoning for Efficient Long-Horizon Agents, 2025. URL https://arxiv.org/abs/2506.15841.

A Evaluation Protocol and Metric Definitions

This appendix specifies the model configuration, evidence units, metric aggregation, and tool execution.

A.1 Model and Retrieval Settings

Table 5: Shared model and retrieval settings.

Item	Setting
Construction / agent / reader	`Qwen/Qwen3.5-27B`; BF16; thinking disabled; temperature 0
Embedding	`Qwen/Qwen3-Embedding-8B`; normalized dense vectors
Reranker	`Qwen/Qwen3-Reranker-8B`
Judge	GPT-5.6 (API name: `gpt-5.6`)
Initial retrieval	BM25 + dense, fused with RRF; RRF constant 60; Top-30 candidates
Initial evidence seeds	At most 10 nodes
Final evidence budget	8,000 tokens
Tool-return text budget	12,000 tokens cumulatively
Traversal limits	Four rounds; at most ten hops per call
Random seeds	13, 37, 71

The open-weight model checkpoints are fixed as follows:

- Qwen3.5-27B: `fc05daec18b0a78c049392ed2e771dde82bdf654`.
- Qwen3-Embedding-8B: `1d8ad4ca9b3dd8059ad90a75d4983776a23d44af`.
- Qwen3-Reranker-8B: `77d193c791ed757ca307ee72715aa132723da912`.

RHELM, LoCoMo, and LongMemEval-S retain their native accuracy, token-level F1, and LLM-judged scoring protocols, respectively. State-query evaluation covers five query types: current state, historical state, reason for change, complete evolution, and final outcome.

A.2 Evidence Recall and Source Cutoffs

Let E_q be the set of gold supporting content for question q, and let D_q^B be the supporting content actually delivered to the reader under budget B. For nonempty E_q,

$$\mathrm{ER@B}(q) = \frac{|E_q \cap D_q^B|}{|E_q|}, \qquad \mathrm{CER@B}(q) = \mathbf{1}[E_q \subseteq D_q^B]. \tag{10}$$

The main retrieval table reports ER@B and CER@B. Source-level ER@10 and CER@10 first deduplicate by source record or session and then retain the first ten items in the final ranking. The initial candidate count, initial seed count, and final source cutoff are recorded separately. Evidence counts as recovered only when its supporting text is included in the final context.

Let L_q and H_q be the evidence sets required for the latest valid state and historical states, respectively. Then

$$\mathrm{LSR}(q) = \frac{|L_q \cap D_q^B|}{|L_q|}, \qquad \mathrm{HR}(q) = \frac{|H_q \cap D_q^B|}{|H_q|}. \tag{11}$$

Each metric uses the query slice with a nonempty corresponding gold set. The strict latest-only control removes overwritten historical states and their accessible copies; the strict historical slice contains only questions that depend on those earlier states.

A.3 Trajectories, Graph Structure, and Staleness

Transition evidence recall measures whether the required endpoints and transition relations are supported by the context. Ordered path exact match requires all necessary states in the correct order; branch recall measures coverage of required branches; temporal-order accuracy measures the ordering of gold state pairs; and outcome evidence support measures direct support for the final state.

Precision and recall of strong and weak graph edges are measured independently of these evidence-recovery metrics.

Staleness detection uses a shared set of true-successor labels. Precision, recall, F1, and the false-positive rate are computed from the confusion matrix:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_1 = \frac{2TP}{2TP + FP + FN}, \quad \mathrm{FPR} = \frac{FP}{FP + TN}. \tag{12}$$

Percentages in the tables equal the corresponding ratios multiplied by 100. Component purity, correct chain assignment, and false merge rate measure matter organization and linking errors.

A.4 Graph Tools and Cost Accounting

Table 6: Inputs and outputs of `DAG_Search`.

Field	Meaning
`unit_id`	Target matter-level state unit
`direction`	`forward`, `backward`, or `both`
`target`	`latest`, `history`, `full_graph`, or `branch`
`start_node_ids`	Starting nodes for traversal
`max_hops`, `budget`	Limits on an individual call
`nodes`, `edges`	Returned state records and relations
`graph_frontier`	Frontier of the returned evidence
`unseen_successor_count`	Number of unobserved successors
`is_stale`, `summary`	Staleness flag and matter summary

Graph calls, agent decision calls, and reader calls are counted separately. The final evidence budget constrains retrieved evidence text; the remaining reader prompt is counted separately. Total online model tokens are

$$T_{\mathrm{online}} = T_{\mathrm{agent,in}} + T_{\mathrm{agent,out}} + T_{\mathrm{reader,in}} + T_{\mathrm{reader,out}}. \tag{13}$$

Here, $T_{\mathrm{reader,in}}$ includes evidence and other input. Tool-return text is counted in the corresponding model input when consumed; text repeated in later inputs is counted for each call. Offline judge cost is reported separately from online query cost. End-to-end latency includes agent decisions and reader generation.

B Complete Experimental Results

This appendix reports the benchmark, ablation, sensitivity, cost, deletion, and human-evaluation results.

B.1 Native Benchmark Results

Each benchmark uses its native scoring protocol.

Table 7: RHELM native question-answering metrics. All scores are percentages.

Condition	Overall accuracy	Temporal accuracy
Hybrid RAG	28	33
RAPTOR	30	37
Mem0	31	40
A-MEM	32	44
MemState	35	52
MemU	33	47
Zep/Graphiti	34	49

Overall accuracy and temporal-subset accuracy are reported separately. Published reference results retain their own evaluation protocols.

Table 8: LoCoMo native token F1. All scores are percentages.

Condition	Single-hop F1	Multi-hop F1	Temporal F1	Open-domain F1
Hybrid RAG	42	28	50	48
RAPTOR	44	30	52	49
Mem0	44	31	53	49
A-MEM	46	33	55	49
MemState	47	35	57	49
Zep/Graphiti	47	34	56	50

All methods use the same version of the benchmark-native token-F1 scoring protocol.

Table 9: LongMemEval-S native accuracy. All scores are percentages.

Condition	Overall accuracy	Update accuracy	Temporal accuracy
Hybrid RAG	82	81	81
RAPTOR	84	83	84
Mem0	85	87	84
A-MEM	86	87	86
MemState	88	92	91
Zep/Graphiti	88	90	89

All methods use LongMemEval-S with the same data split and answer-scoring protocol.

B.2 State Evidence and Trajectory Recovery

State-evidence metrics, scores for the five question types, and trajectory metrics use the same method configurations.

Table 10: State evidence and answer quality under the shared protocol. Scores are percentages; graph calls are averages per question.

Condition	ER@B	CER@B	Latest recall	History recall	State QA	Graph calls/q.
Hybrid RAG	70	49	68	61	70	0
RAPTOR	74	55	74	66	74	0
Mem0	77	60	86	49	75	0
A-MEM	80	65	83	72	79	0
MemState	88	76	93	85	86	2.2

ER and CER are computed from supporting text in the final context. State QA uses the shared scoring protocol.

Table 11: Answer quality for the five state-question types. All scores are percentages.

Condition	Current QA	History QA	Reason QA	Evolution QA	Outcome QA	State QA
Hybrid RAG	79	67	64	56	84	70
RAPTOR	82	73	69	61	85	74
Mem0	93	64	68	60	90	75
A-MEM	89	78	73	66	89	79
MemState	94	88	83	75	90	86

The five question types are scored separately. State QA is macro-averaged across question types and datasets.

Table 12: Transition, path, branch, and outcome evidence. All scores are percentages.

Condition	Transition recall	Ordered path EM	Branch recall	Temporal order acc.	Outcome support
Hybrid RAG	47	28	35	49	74
RAPTOR	51	32	40	53	77
Mem0	33	18	25	40	84
A-MEM	63	44	51	64	82
MemState	83	67	74	81	91

Transition, path, branch, order, and final-outcome metrics are evaluated against independently annotated supporting content.

B.3 State Preservation, Graph Structure, and Retrieval Decisions

Strict state-deletion controls, graph-structure factorial experiments, and compute controls separately test accessible history, connection structure, and adaptive retrieval.

Table 13: Strict state preservation $\times$ graph traversal controls. Scores are percentages; graph calls are averages per question.

Condition	ER@B	CER@B	Latest recall	History recall	State QA	Graph calls/q.
Strict latest-only + static retrieval	46	34	86	0	39	0
Hybrid RAG	70	49	68	61	70	0
Strict latest-only + ASGR	46	34	86	0	39	0
MemState	88	76	93	85	86	2.2

The strict latest-only condition removes overwritten historical states and their accessible copies. The two latest-only controls use the same reachable evidence. CER is evaluated on the mixed question set.

Table 14: Graph structure $\times$ traversal factorial experiment. All scores are percentages.

Condition	ER@B	CER@B	Latest recall	History recall	State QA
Local temporal chain + static RAG	70	49	68	61	70
DMSG + static RAG	70	49	68	61	70
Local temporal chain + ASGR	84	69	89	78	81
DMSG + ASGR	88	76	93	85	86

The two graph structures are each evaluated with static retrieval and ASGR. The static conditions use identical records and rankings.

Table 15: Tool rounds under the same evidence-budget limits. Scores are percentages; graph calls are averages per question.

Condition	ER@B	CER@B	Latest recall	History recall	State QA	Graph calls/q.
No traversal (0 rounds)	70	49	68	61	70	0
Up to 1 round	82	65	85	75	80	1
Up to 2 rounds	86	73	91	82	84	1.7
Full ASGR (up to 4 rounds)	88	76	93	85	86	2.2
Up to 6 rounds	89	77	93	86	86	2.6
Up to 8 rounds	89	77	93	86	85.5	2.9

Maximum rounds are 0, 1, 2, 4, 6, and 8. The final evidence budget is 8K tokens, and cumulative tool-returned text is capped at 12K tokens.

Table 16: Compute controls and adaptive decisions. Scores are percentages; graph calls are averages per question.

Condition	ER@B	CER@B	Latest recall	History recall	State QA	Graph calls/q.
Compute-matched multi-round text search	84	70	88	77	82	0
Fixed-direction graph expansion	85	71	89	79	82	2.2
MemState	88	76	93	85	86	2.2

This comparison includes multi-round text search, fixed-direction graph expansion, and adaptive graph retrieval. Graph calls are counted separately.

B.4 Construction Quality and Frontier Decisions

We evaluate relation classification, chain assignment, false merges, and staleness detection.

Table 17: DMSG construction quality and downstream performance. All scores are percentages.

Condition	Strong precision	Weak precision	Strong recall	Component purity	Correct chain assignment	History recall	State QA
MemState	93	88	89	91	83	85	86
Without weak contextual edges	93	N/A	89	91	83	77	82
Untyped relations	N/A	N/A	N/A	84	71	80	81
Without chain-level summaries	86	81	85	82	60	79	80
Without summary-consistency rejection	88	82	92	85	69	81	81
Summary-only matching	90	85	86	86	66	79	81
Local-node matching only	84	80	90	79	53	74	77

The no-weak-edge condition preserves the strong-edge structure. Strong/weak relation-classification metrics are not applicable when relations are untyped. Chain assignment is evaluated on the 32-hop slice.

Table 18: Effect of DMSG on trajectory recovery. All scores are percentages.

Condition	Transition recall	Ordered path EM	Branch recall	Outcome support
MemState	83	67	74	91
Without weak contextual edges	82	64	67	88
Untyped relations	75	58	65	87
Without chain-level summaries	72	55	64	86
Without summary-consistency rejection	75	58	66	86
Summary-only matching	74	57	65	87
Local-node matching only	67	47	57	83

Construction variants are evaluated on the same trajectory-question set.

Table 19: Long-chain assignment stress test. All scores are percentages.

Condition	Correct chain assignment
1 hop / local	96
1 hop / summary	96
1 hop / full	97
2 hops / local	94
2 hops / summary	94
2 hops / full	96
4 hops / local	88
4 hops / summary	91
4 hops / full	94
8 hops / local	80
8 hops / summary	86
8 hops / full	92
16 hops / local	67
16 hops / summary	77
16 hops / full	88
32 hops / local	53
32 hops / summary	66
32 hops / full	83

Matter-assignment accuracy is stratified by the number of hops in the evolution chain.

Table 20: False merges and true links for distinct matters with overlapping wording. TP, FP, FN, and TN are counts; merge rate and link recall are percentages.

Condition	TP	FP	FN	TN	False-merge rate	True-link recall
Embedding threshold	356	108	44	492	18	89
Local LLM judgment	348	78	52	522	13	87
Summary matching	348	60	52	540	10	87
Full DMSG	352	42	48	558	7	88

TP, FP, FN, and TN count positive and negative link decisions. False-merge rate and true-link recall are reported separately.

Table 21: Frontier staleness detection. TP, FP, FN, and TN are counts; all other columns are percentages.

Condition	TP	FP	FN	TN	Precision	Recall	F1	FPR	State QA
Timestamp only	280	150	120	450	65.1163	70	67.4699	25	77
Newest retrieved node	332	84	68	516	79.8077	83	81.3725	14	83
Retrieved frontier	352	48	48	552	88	88	88	8	86
Oracle successor labels	400	0	0	600	100	100	100	0	89

For each condition, precision, recall, F1, and FPR are computed from its TP, FP, FN, and TN counts; $\mathrm{FPR} = \mathrm{FP}/(\mathrm{FP} + \mathrm{TN})$.

B.5 Budgets, Perturbations, and Generalization

We evaluate sensitivity to evidence budgets, graph and timestamp perturbations, update density, memory scale, reader models, and initial retrieval backends.

Table 22: Final evidence-context budget. All scores are percentages.

Condition	CER@B	History recall	State QA
2K final evidence tokens	55	64	75
4K final evidence tokens	66	76	81
8K final evidence tokens	76	85	86
16K final evidence tokens	79	88	87

The final evidence-context budgets are 2K, 4K, 8K, and 16K tokens.

Table 23: Number of initial candidates. All scores are percentages.

Condition	CER@B	History recall	State QA
Top-10 candidates	65	74	80
Top-20 candidates	72	82	84
Top-30 candidates	76	85	86
Top-50 candidates	77	85	86

The initial candidate count varies while the final evidence budget and traversal policy remain fixed.

Table 24: Graph-structure perturbations. All scores are percentages.

Condition	CER@B	State QA
MemState	76	86
Randomly remove 5% of strong edges	73	84
Randomly remove 10% of strong edges	69	81
Randomly remove 20% of strong edges	62	76
Add 5% incorrect strong edges	70	81
Add 10% incorrect strong edges	63	76

Strong edges are deleted or incorrect strong edges are added while preserving the directed acyclic constraint.

Table 25: Timestamp-field perturbations. All scores are percentages.

Condition	Latest recall	State QA
MemState	93	86
Timestamp noise ± 1 day	92	85
Timestamp noise ± 7 days	88	81
Remove 20% of timestamp fields	86	80
Remove all retrieval timestamp fields	78	74

Timestamp fields are perturbed after graph construction and before retrieval; graph structure remains fixed.

Table 26: Results by update density. All scores are percentages.

Condition	State QA
Hybrid / 0 updates	83
Hybrid / 1 update	77
Hybrid / 2-3 updates	69
Hybrid / 4-7 updates	59
Hybrid / 8+ updates	47
Mem0 / 0 updates	82
Mem0 / 1 update	80
Mem0 / 2-3 updates	75
Mem0 / 4-7 updates	66
Mem0 / 8+ updates	55
A-MEM / 0 updates	84
A-MEM / 1 update	82
A-MEM / 2-3 updates	77
A-MEM / 4-7 updates	70
A-MEM / 8+ updates	61
MemState / 0 updates	84
MemState / 1 update	85
MemState / 2-3 updates	86
MemState / 4-7 updates	84
MemState / 8+ updates	80

State QA is grouped by the number of updates to the same matter.

Table 27: Memory scaling. All scores are percentages.

Condition	Latest recall
Hybrid / 1000 records	79
Mem0 / 1000 records	88
MemState / 1000 records	94
Hybrid / 5000 records	75
Mem0 / 5000 records	86
MemState / 5000 records	94
Hybrid / 10000 records	72
Mem0 / 10000 records	84
MemState / 10000 records	93
Hybrid / 50000 records	63
Mem0 / 50000 records	79
MemState / 50000 records	91
Hybrid / 100000 records	58
Mem0 / 100000 records	74
MemState / 100000 records	89

The target matter and required evidence remain fixed; memory scale increases by adding distracting memories.

Table 28: Reader capacity with fixed retrieved evidence. All scores are percentages.

Condition	State QA
Qwen/Qwen3.5-9B / Mem0	65
Qwen/Qwen3.5-9B / MemState	78
Qwen/Qwen3.5-27B / Mem0	75
Qwen/Qwen3.5-27B / MemState	86
GPT-5.6 / Mem0	83
GPT-5.6 / MemState	89

Graph construction and retrieved evidence remain fixed; only the reader changes.

Table 29: Initial retrieval backends and graph expansion. All scores are percentages.

Condition	ER@B
BM25 / Static	60
BM25 / ASGR	82
Qwen/Qwen3-Embedding-0.6B / Static	68
Qwen/Qwen3-Embedding-0.6B / ASGR	84
Qwen/Qwen3-Embedding-4B / Static	69
Qwen/Qwen3-Embedding-4B / ASGR	87
Qwen/Qwen3-Embedding-8B / Static	70
Qwen/Qwen3-Embedding-8B / ASGR	88

The reranker remains fixed. BM25-only retrieval and hybrid retrieval with different Qwen encoders are evaluated separately.

Table 30: Metrics at a cutoff of ten in the final source ranking. All scores are percentages.

Condition	Source ER@10	Source CER@10
Hybrid RAG	68	45
RAPTOR	71	50
Mem0	74	54
A-MEM	77	59
MemState	85	70

The final source ranking is deduplicated and truncated to ten sources. These metrics are reported separately from metrics over the full evidence context.

B.6 Cost, Deletion, and Human Evaluation

We report online and construction costs, deletion outcomes, and human annotation agreement.

Table 31: Online token accounting for quality–cost comparison. All entries are token counts.

Condition	Final evidence	Reader other in.	Reader output	Agent total in.	Agent total out.	Tool text total	Online total
Hybrid RAG	7600	500	300	0	0	0	8400
RAPTOR	7600	500	300	0	0	0	8400
Mem0	6600	500	300	0	0	0	7400
A-MEM	7600	500	300	0	0	0	8400
ASGR: up to 1 round	7600	500	300	4500	240	2200	13140
ASGR: up to 2 rounds	7600	500	300	7500	400	4000	16300
ASGR: up to 4 rounds	7600	500	300	12000	600	5200	21000
ASGR: up to 6 rounds	7600	500	300	14500	700	6100	23600
ASGR: up to 8 rounds	7600	500	300	16200	780	6500	25380
Compute-matched multi-round text search	7600	500	300	12000	600	5200	21000

The online total sums agent input and output with reader input and output. Tool-returned text is not counted a second time; offline judging is accounted for separately.

Table 32: Call counts and end-to-end latency. Calls are averages per question; latency is in seconds.

Condition	Graph calls/q.	Agent calls/q.	Online LLM calls/q.	Online p50 (s)	Online p95 (s)
Hybrid RAG	0	0	1	10	25
RAPTOR	0	0	1	10	25
Mem0	0	0	1	9	23
A-MEM	0	0	1	10	25
ASGR: up to 1 round	1	2	3	21	50
ASGR: up to 2 rounds	1.7	2.7	3.7	29	67
ASGR: up to 4 rounds	2.2	3.2	4.2	36	80
ASGR: up to 6 rounds	2.6	3.6	4.6	42	96
ASGR: up to 8 rounds	2.9	3.9	4.9	47	110
Compute-matched multi-round text search	0	3.2	4.2	36	80

Graph, agent, and reader calls are counted separately. Online end-to-end latency excludes the offline judge.

Table 33: Construction overhead per new memory record. Calls and tokens are measured per new memory record.

Condition	Calls /record	Input tokens /record	Output tokens /record	Total tokens /record
Hybrid RAG	0	0	0	0
RAPTOR (amortized)	0.2	400	100	500
Mem0	2	1800	300	2100
A-MEM	3	2600	500	3100
DMSG	3	3000	600	3600

Construction calls and token costs are measured per new memory record.

Table 34: Offline scoring cost. All entries are token counts.

Condition	Judge input	Judge output	Judge total
GPT-5.6 offline scoring	1000	100	1100

Offline-judge input, output, and total tokens are accounted for independently.

Table 35: Deletion propagation and residual information. Failed requests and leaking queries are counts; the three rates are percentages.

Condition	Failed requests	Leaking queries	Request residual	Request pass rate	Query residual
Source records only	36	90	18	82	9
Source records + derived edges	12	20	6	94	2
Full cascade (index, summaries, caches)	1	1	0.5	99.5	0.1
TTL with full cascade	2	3	1	99	0.3

Request-level deletion failure and pass rates are complementary. Query-level residual information is measured separately.

Table 36: Human annotation agreement. TP, FP, FN, and TN are paired-label counts; agreement is in percent and κ is dimensionless.

Condition	TP	FP	FN	TN	Agreement	Cohen's κ
Human evidence support (binary)	126	21	24	129	85	0.7

The counts represent the paired annotators' binary evidence-support labels, together with agreement and Cohen's kappa. The four cells denote both positive, first positive/second negative, first negative/second positive, and both negative, respectively.

C Published Reference Results and Evaluation Details

This appendix provides published benchmark references, metric aggregation rules, and application scope.

C.1 Published Benchmark References

RHELM reference values use the official configuration without external data sources (Zhang et al., 2026). LoCoMo reference results come from the A-MEM and Mem0 papers (Xu et al., 2025; Chhikara et al., 2025). LongMemEval-S reference values correspond to Zep GPT-4o and Mem0 Platform Top-50. Sources include the `hindsight-benchmarks` and `memory-benchmarks` repositories.[1] Direct method comparisons in this paper use the unified results in the main text and Appendix B.

Table 37: RHELM QA accuracy. FC, TP, AG, HL, MI, EX, and MX denote fact, temporal, aggregation, hallucination, misleading, external-source, and mixed categories. Avg is the reported overall score.

Method	FC	TP	AG	HL	MI	EX	MX	Avg
Hybrid RAG k=20 (public)	47.3	35.7	31.8	19.3	3.1	10.4	15.2	24.8
Mem0 (public)	41.6	31.4	28.1	10.7	3.1	10.8	13.3	21.1
MemU (public)	49.3	32.4	33.9	8.6	4.6	12.0	11.4	23.1

Table 38: LoCoMo token-level F1. Public reference rows retain the source papers' evaluation protocols.

Method	Single-hop	Multi-hop	Temporal	Open-domain
A-MEM (public)	44.65	27.02	45.85	12.14
Mem0 (public)	38.72	28.64	48.93	47.65
Mem0^g (public)	38.09	24.32	51.55	49.27

[1] `https://github.com/vectorize-io/hindsight-benchmarks` and `https://github.com/mem0ai/memory-benchmarks`.

Table 39: LongMemEval-S accuracy. SS denotes single-session categories. Mem0 Platform uses the public Top-50 setting.

Method	SS User	SS Assistant	SS Preference	Knowledge update	Temporal	Multi-session	Overall
Zep GPT-4o (public)	92.9	80.4	56.7	83.3	62.4	57.9	71.2
Mem0 Platform (public)	98.6	98.2	93.3	93.6	94.0	93.2	94.8

C.2 Aggregation and Human Evaluation

State QA uses a shared binary scoring rule, first macro-averaging over query types within each dataset and then macro-averaging over datasets. Native benchmark accuracy and token-level F1 are reported separately.

Human agreement on evidence support uses the joint binary labels from two annotators. The four counts in Table 36 are 126, 21, 24, and 129, totaling 300 examples; agreement is 85.0% and Cohen's κ is 0.70. Request-level deletion pass rate, request-level residual rate, and query-level residual rate each use their corresponding denominators.

C.3 Application Scope

MemState addresses continually changing matters and queries about their history, reasons for change, intermediate states, and final outcomes. Graph transitions describe state progression between memory records. Construction, source tracing, and retrieval tools jointly support evidence-grounded answers; authorization, retention periods, and deletion policies are specified by the application.