FROM ISOLATED FACTS TO MEMORY TRAJECTORIES: LIGHTWEIGHT RETRIEVAL FOR DYNAMICALLY EVOLVING CONVERSATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon conversations do not preserve a user state as independent facts: evidence is distributed across sessions, revised over time, and often relevant only through intermediate memories. Existing memory systems recover these dependencies via costly mechanisms: graph-structured memories construct elaborate relational structures during indexing and rely on multi-stage graph traversal at query time, while agentic retrievers repeatedly invoke LLMs to plan and refine retrieval. We propose **TraceMem**, a lightweight alternative that organizes conversational histories into an evolving flat memory. At retrieval time, TraceMem uses two complementary relations: the intra-relation links historical and current expressions of the same memory to preserve state evolution, while the inter-relation connects memories through bridge entities to recover cross-session dependencies. Together they form complete memory trajectories, from which multi-hop and temporal questions are answered without complex graph reasoning or LLM-guided iterative retrieval. Across LoCoMo-10 and LongMemEval-S, TraceMem outperforms agentic baselines by 6.1% while achieving near-zero-latency retrieval, demonstrating that structured evidence reconstruction does not require a complex query-time reasoning system.

1 INTRODUCTION

Conversational question answering (CQA) often asks an agent to answer questions using information accumulated throughout an ongoing dialogue, rather than relying only on general world knowledge. In long-horizon interactions, this dialogue may span months and many sessions, while the agent can examine only a finite context at any one time (Wang et al., 2023; Du et al., 2025; Maharana et al., 2024). This creates a fundamental mismatch: the interaction history grows without bound, but the memory supplied to the reader must remain compact. The difficulty is also more than length. A user's state changes as plans are completed, preferences shift, and earlier statements are corrected; the question and its supporting dialogue may use different words; and answering often requires ordering events in time, combining evidence from separate sessions, or identifying which of several conflicting statements is current (Wu et al., 2025; Chhikara et al., 2025). Long-horizon CQA is therefore not simply the retrieval of a similar utterance, but the reconstruction of a small, temporally coherent body of evidence from a noisy and continuously evolving history.

Existing long-term memory systems adopt multiple approaches to recover these non-local dependencies. Some materialize multi-granularity associations or explicit relational graphs, then exploit those structures during retrieval (Xu et al., 2026b; Van et al., 2026; Jiang et al., 2026). Their expressivity can require costly offline extraction, entity resolution, relation construction, and index maintenance, followed by non-trivial graph-guided retrieval. Others use LLM calls to construct, update, consolidate, or route stored experience (Xu et al., 2025; Chhikara et al., 2025; Kang et al., 2025; Liu et al., 2026); more elaborate variants interleave planning with repeated tool-based retrieval (Banerjee et al., 2026; Wang et al., 2026). Consequently, substantial token and latency costs may arise during construction, retrieval, or both. The problem becomes more pronounced as histories evolve: a new or contradictory observation may invalidate derived summaries, links, clusters, or graph properties and require extensive rewriting or a complete reindexing process.

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.

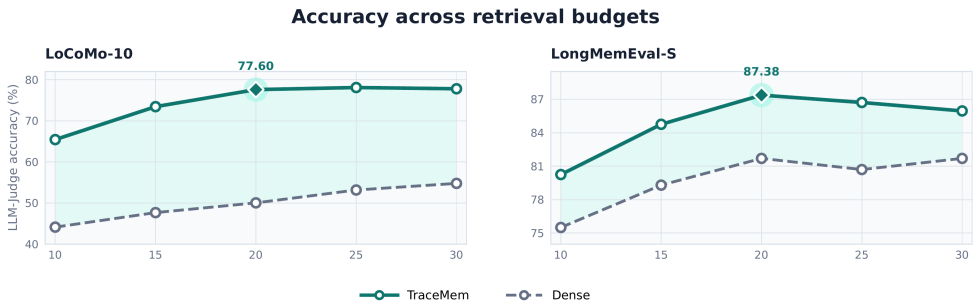

Figure 1: Answer accuracy peaks at an effective context length. LLM-Judge accuracy is reported across top-K source-turn budgets on LoCoMo-10 and LongMemEval-S.

Our central insight is that the evidence needed for conversational QA can be assembled without complex graph reasoning or an online retrieval agent. TraceMem stores dialogue content as propositions and links them through only two complementary relations. The *intra-relation* connects historical and current expressions of the same underlying memory, preserving how a state evolves and which version is current; the *inter-relation* connects otherwise distinct memories through bridge entities, exposing dependencies that cross topics and sessions. At query time, these relations assemble a compact *memory trajectory* whose propositions help aggregate the relevant evidence required by the reader, even when they are not directly connected by semantic similarity. This formulation turns retrieval into lightweight, deterministic scoring over flat records while retaining the context needed for questions whose answers depend on a single fact, a chain of facts, temporal order, or successive revisions of user state.

This design makes better use of a fixed reader budget. As shown in Figure 1, TraceMem consistently outperforms direct dense retrieval under the same source-turn budget on both benchmarks. At $K = 20$, both methods supply exactly 20 turns to the same reader, yet TraceMem connects complementary propositions into trajectories before mapping their collective evidence back to the dialogue. Dense retrieval ranks each raw turn independently and can miss supporting evidence whose relevance emerges only through another memory; the resulting gap therefore comes from evidence assembly, not additional context.

Our contributions are threefold:

1. **Memory-efficient evolving store.** We introduce a flat proposition memory that incrementally incorporates a continually growing dialogue history while retaining the lineage needed to resolve evolving and conflicting user states.
2. **Memory-trajectory retrieval.** We formulate retrieval through intra- and inter-relations that assemble complete evidence trajectories for both single-hop and multi-hop conversational QA, including temporal and cross-session questions.
3. **Performance and efficiency.** TraceMem establishes a state-of-the-art performance–efficiency trade-off across long-horizon CQA benchmarks, achieving up to a 6.1% relative improvement over a strong agentic baseline while substantially reducing online context and retrieval latency.

2 Related Work

Structured and graph-guided memory. Structured memories organize records beyond an independent flat collection, using retrieval trees, knowledge graphs, or multi-granular structures to compose dispersed evidence (Sarthi et al., 2024; Gutiérrez et al., 2024; Ke et al., 2025; Hieu et al., 2025). RAPTOR recursively summarizes clustered passages, whereas HippoRAG propagates relevance over an induced knowledge graph. MemGAS selects among memory granularities, MemO-RAI retrieves provenance-aware subgraphs, and MAGMA routes queries across several typed graphs (Xu et al., 2026b; Van et al., 2026; Jiang et al., 2026). StructMem periodically consolidates temporally anchored events, while APEX-MEM couples a temporal property graph with tool-based retrieval (Xu et al., 2026a; Banerjee et al., 2026). Across these variants, non-local evidence is made

explicit in persistent nodes and edges, so new sessions can require relational structures to be revised before later queries traverse them. These designs recover non-local dependencies but add structure construction, maintenance, or traversal; TraceMem instead links flat propositions through lineage and bridge entities.

LLM-orchestrated memory. Other systems delegate memory organization or access to pretrained LLMs. A-Mem evolves linked notes, Mem0 applies model-guided memory operations, and MemoryOS promotes information through hierarchical tiers (Xu et al., 2025; Chhikara et al., 2025; Kang et al., 2025). LightMem combines topic-aware short-term storage with parallel sleep-time consolidation, whereas EverMemOS builds MemCells and thematic MemScenes before agentic recollection (Fang et al., 2026; Hu et al., 2026). SimpleMem and reflective memory management combine compression with retrieval planning; E-Mem extends this pattern with a planner and episodic assistant agents (Liu et al., 2026; Tan et al., 2025; Wang et al., 2026; Trivedi et al., 2023). These systems offer flexible memory control, but planning, reflection, or agent coordination introduces additional model-dependent steps before the final evidence reaches the reader. TraceMem keeps model-dependent transformations at indexing while using deterministic trajectory scoring during inference.

3 METHODOLOGY

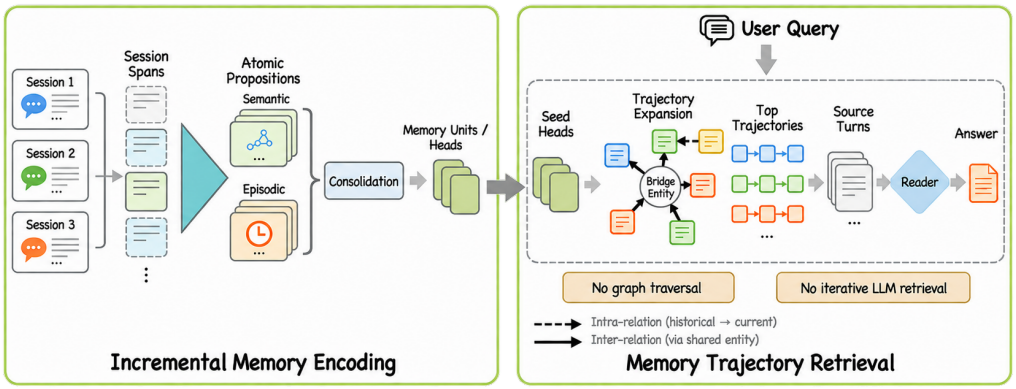

Figure 2: Overview of TraceMem. **Left: Incremental Memory Encoding** organizes incoming sessions into coherent spans, forms semantic and episodic propositions, and consolidates them into evolving memory heads. **Right: Memory Trajectory Retrieval** selects seed heads, expands trajectories through intra- and inter-relations, maps the top trajectories to grounded source turns, and invokes the reader once, without complex graph reasoning or iterative LLM retrieval.

3.1 PRELIMINARIES

Motivation. Long-horizon conversational QA is not merely a nearest-neighbor lookup problem. A question may refer to a recent state using words that appeared only in an earlier session, or require evidence scattered across sessions whose individual pieces are only weakly related to the question. A passive top-k retriever scores every memory independently against the original query and can therefore miss a supporting fact that becomes relevant only after another fact has been found. Agentic retrieval can adapt the search after each intermediate result, but usually does so through repeated query reformulation and LLM calls. TraceMem seeks the same evidence-conditioned behavior through a lightweight retrieval process over an incrementally updated memory.

Problem formulation. Let a conversation be an ordered collection of sessions $\mathcal{C} = (\mathcal{S}_1, \ldots, \mathcal{S}_N)$, where session $\mathcal{S}_i = (t_{i,1}, \ldots, t_{i,T_i})$ contains a sequence of dialogue turns. Sessions arrive sequentially and update a persistent proposition memory $\mathcal{M}_i$ together with its current-head view $\mathcal{H}_i \subseteq \mathcal{M}_i$,

$$(\mathcal{M}_0, \mathcal{H}_0) = (\varnothing, \varnothing), \qquad (\mathcal{M}_i, \mathcal{H}_i) = \mathcal{I}((\mathcal{M}_{i-1}, \mathcal{H}_{i-1}), \mathcal{S}_i), \quad i = 1, \ldots, N, \tag{1}$$

where $\mathcal{I}$ denotes the indexing and consolidation process described in §3.2. Given a question q, retrieval begins with a small set of seed heads selected from $\mathcal{H}_N$. Starting from each seed C_1, TraceMem appends supporting heads $C_2, \ldots, C_d$ that are reachable through normalized or equivalent bridge entities. This produces a memory trajectory $\pi = (C_1, \ldots, C_d)$. Let $\Pi^\star$ denote the final trajectories returned by the procedure in §3.3. They are mapped back to their source dialogue turns before answering:

$$\mathcal{T}_q = \Gamma(\Pi^\star), \qquad \hat{y} = \mathbb{G}(q, \Pi^\star, \mathcal{T}_q). \tag{2}$$

Here, Γ recovers source turns from the trajectory lineages and $\mathbb{G}$ is the reader. Propositions thus guide evidence aggregation, while the original turns in $\mathcal{T}_q$ provide the grounded evidence used to answer the question. Mapping each trajectory back to the original turns provides a common grounding interface for controlled comparison with other memory systems; the trajectory heads are auxiliary representations constructed within TraceMem, so no externally supplied memory units are required.

A Cognitive Perspective. TraceMem views retrieval through the lens of reconstructive recall. Neuroscience-inspired retrieval work models a partial cue as reactivating an index that reinstates distributed components of an episode (Gutiérrez et al., 2024). In TraceMem, the question serves as this partial cue and a retrieved seed as the reactivated index: once found, it exposes bridge entities through which additional evidence can be reinstated, even when that evidence is only weakly related to the original question. This gives rise to *active retrieval*, where what has already been recalled determines what becomes accessible next. Recent analyses likewise show that reinstated temporal context can govern episodic-like order recall in long-context language models (Pink et al., 2026). A *memory trajectory* carries this context forward: every recovered proposition both contributes evidence and reshapes the context from which the next proposition is recalled, until the complete trajectory forms a reconstructed recollection that can be evaluated as a whole. We elaborate how TraceMem operationalizes these two principles in the trajectory retrieval procedure of §3.3.

3.2 Incremental Memory Encoding

As illustrated on the left of Figure 2, TraceMem encodes each incoming session in three write-time stages: it organizes the dialogue into coherent spans, distills self-contained semantic and episodic propositions, and consolidates them into the persistent memory. These stages run only as the conversation grows and are never repeated for a query.

Abstract Layer. The Abstract Layer provides a transient write-time view of a session as locally coherent spans. This intermediate organization is useful because a session may shift among plans, experiences, and personal updates, whereas individual turns often depend on nearby context to resolve their meaning. Accordingly, the layer partitions each session $\mathcal{S}_i$ into a chronologically ordered collection of contiguous spans:

$$\begin{gathered} \mathcal{Z}_i = \mathcal{F}_\theta(\mathcal{S}_i) = \{z_{i,1}, \ldots, z_{i,J_i}\}, \\ \bigcup_{j=1}^{J_i} z_{i,j} = \mathcal{S}_i, \qquad z_{i,j} \cap z_{i,k} = \varnothing \ (j \neq k), \qquad z_{i,j} \prec z_{i,j+1}. \end{gathered} \tag{3}$$

The spans are disjoint and jointly cover the session, as encoded above. Each inherits the session date d_i and its source-turn identifiers, providing the local context needed to form self-contained atomic representations and cleaner intra-relations. They remain temporary scaffolds and are neither stored in persistent memory nor used in query-time scoring.

Atomic Layer. The Atomic Layer is the persistent representation that transforms each coherent span into self-contained semantic and episodic propositions, together with their salient entities, temporal scope, and source-turn provenance. Earlier turns from the same session may provide the context needed to interpret omitted or ambiguous content, while supporting evidence remains restricted to the current span. Thus, we write this unified transformation as

$$\begin{aligned} \big(\mathcal{P}_{i,j}, \{\mathcal{E}(p)\}_{p \in \mathcal{P}_{i,j}}\big) &= \mathcal{F}_\theta(z_{i,j}; d_i) \\ &\approx \big((g_{\text{layer}}, g_{\text{entity}}) \circ g_{\text{ground}} \circ g_{\text{compress}}\big)(z_{i,j}; d_i), \\ \mathcal{P}_{i,j} &= \mathcal{P}^{\text{sem}}_{i,j} \cup \mathcal{P}^{\text{epi}}_{i,j}. \end{aligned} \tag{4}$$

Within this pass, $\mathcal{F}_\theta$ follows four constraints: (1) replace ambiguous references with explicit entities, (2) ground relative temporal expressions to absolute dates using d_i, (3) decompose compound dialogue into self-contained factual propositions, and (4) extract the salient people, objects, places, and events associated with each proposition. The first two are captured by g_{ground}, the third by g_{compress}, while g_{layer} separates relatively stable knowledge from time-bound experiences and g_{entity} produces the entity set $\mathcal{E}(p)$. The extracted surface forms are then deterministically canonicalized by Unicode and case normalization. Every proposition retains these entities together with its temporal scope and source-turn provenance. Collecting $\mathcal{P}_{i,j}$ across all spans yields the session-level content subsequently consolidated into $\mathcal{M}$.

Semantic Propositions. For each span $z_{i,j}$, the subset $\mathcal{P}^{\text{sem}}_{i,j} \subseteq \mathcal{P}_{i,j}$ captures knowledge whose validity extends beyond a particular event. It includes personal attributes, preferences, relationships, recurring habits, skills, background knowledge, and general facts. Each member of $\mathcal{P}^{\text{sem}}_{i,j}$ states one context-independent claim while retaining the turns in $z_{i,j}$ that support it. These propositions are usually not anchored to the session date d_i; however, an explicit validity period is preserved whenever it changes the meaning of the claim. As later sessions contribute new semantic propositions, compatible evidence can refine the current representation, whereas revised or conflicting knowledge remains connected through consolidation lineage. In this way, the semantic subset provides a compact but traceable account of the user's evolving knowledge state.

Episodic Propositions. By contrast, $\mathcal{P}^{\text{epi}}_{i,j} \subseteq \mathcal{P}_{i,j}$ captures what happened and when, preserving events, plans, decisions, outcomes, and state transitions as time-bound observations. Each episodic proposition retains the participants and salient circumstances supported by $z_{i,j}$, such as the object, location, motivation, or consequence of an event. Its temporal expressions are grounded against d_i, with relative references normalized to absolute dates whenever possible. This event-like representation allows related observations to be connected across sessions even when they use different expressions or reveal different aspects of the same episode. Consequently, $\mathcal{P}^{\text{epi}}_{i,j}$ preserves the chronological evidence required for temporal, cross-session, and evolving-state reasoning, while $\mathcal{P}^{\text{sem}}_{i,j}$ summarizes the state that persists across those events.

Consolidation. The atomic propositions are written to $\mathcal{M}$ in conversation order. Within the session-level transformation $\mathcal{I}$, let n index proposition writes, let p be the next proposition, let $e(p)$ be its unit-normalized embedding, and let $\mathcal{H}_n$ contain the current heads after n writes. TraceMem first finds up to k_c related heads, ranked by the displayed lineage maximum, allowing either a head or any of its earlier versions to provide the match:

$$\mathcal{H}_p = \text{Top}_{k_c}\left\{ C \in \mathcal{H}_n : \max_{p' \rightsquigarrow C} \cos\big(e(p), e(p')\big) > \eta \right\}, \qquad (a, \mathcal{H}'_p, C_p) = \mathcal{F}_\theta(p, \mathcal{H}_p). \tag{5}$$

Here, $p' \rightsquigarrow C$ means that the lineage of p' ends at the current head C; we use the relation reflexively, so $C \rightsquigarrow C$. The subset $\mathcal{H}'_p \subseteq \mathcal{H}_p$ contains the heads selected for the update. If $\mathcal{H}_p$ is empty, p is stored directly and no transformation is needed. Otherwise, $\mathcal{F}_\theta$ compares p with these related heads and selects one of three actions: noop keeps p separate; add combines complementary details into a consolidated head; and sup makes p the current representation when it updates, conflicts with, or renders previous heads redundant. The corresponding update acts jointly on the persistent memory and its active heads:

$$(\mathcal{M}_{n+1}, \mathcal{H}_{n+1}) = \begin{cases} (\mathcal{M}_n \cup \{p\},\ \mathcal{H}_n \cup \{p\}), & a = \textsf{noop}, \\ \big(\mathcal{M}_n \cup \{p, C_p\},\ (\mathcal{H}_n \setminus \mathcal{H}'_p) \cup \{C_p\}\big), & a \in \{\textsf{add}, \textsf{sup}\}. \end{cases} \tag{6}$$

Here, C_p denotes the new current head. Under add it consolidates complementary information, whereas under sup it expresses the incoming state while inheriting the entities and source turns of $\mathcal{H}'_p$. In both cases, the selected heads and p remain in $\mathcal{M}$ as historical members whose lineages end at C_p: for every $C \in \mathcal{H}'_p$, all $p' \rightsquigarrow C$ and the incoming p subsequently satisfy $p' \rightsquigarrow C_p$ and $p \rightsquigarrow C_p$. Thus, only the active view $\mathcal{H}$ is replaced; no proposition or provenance is removed from persistent memory. Redundant heads are absorbed by the same transition, so no fourth action is needed. Finally, each update type may occur at most L times along one lineage; an action beyond this budget becomes noop to limit repeated rewriting and semantic drift.

3.3 MEMORY TRAJECTORY RETRIEVAL

Active Retrieval over Passive Retrieval. The right side of Figure 2 summarizes the query-time procedure. Retrieval operates on a fixed snapshot: let $\mathcal{M} = \mathcal{M}_n$ and $\mathcal{H} = \mathcal{H}_n$ denote the latest memory and its current heads when the query arrives. A passive retriever fixes the relevance of every head from the question alone and commits to all of its selections at once. Active retrieval instead lets the evidence accumulated before depth d determine what can be recovered next:

$$\begin{aligned} \mathcal{R}^k_{\mathrm{pas}}(q) &= \mathrm{Top}_k\big\{C \in \mathcal{H} : \bar{s}(q, C)\big\}, \\ C_1 = \mathcal{R}_1(q), \qquad C_d &= \mathcal{R}_d(q, C_{<d}), \qquad C_{<d} = (C_1, \ldots, C_{d-1}). \end{aligned} \tag{7}$$

TraceMem realizes the latter behavior without an online reasoning agent: q is used to locate the first heads, whereas the propositions already collected expose the bridge entities and retrieval context used at subsequent depths.

Proposition 1 (Active retrieval over passive retrieval)**.** *Consider a bridge tree of depth D and branching factor $b \geq 2$. The question identifies only the root, each selected non-leaf head reveals the unique bridge leading to the next correct head, and the answer appears only at the resulting leaf. An evidence-conditioned retriever reaches that leaf while retaining D heads. If the answer leaf is uniform among the b^{D-1} leaves, any query-only passive retriever with k retained-head slots succeeds with probability at most $\min\{1, k/b^{D-1}\}$ and requires $k \geq b^{D-1}$ to guarantee recovery.*

Proposition 1 isolates the relevant advantage: evidence that is uninformative under the original question may become uniquely accessible after an earlier proposition reveals the appropriate bridge.

Intra-relation Links. A conversational fact may be revised over time or simply restated in different words. Consolidation preserves these observations in a lineage $p' \rightsquigarrow C$, where C is its current head. At retrieval, TraceMem compares q with every proposition in memory. The score of a head is then inherited from the best-matching proposition in its lineage, and the strongest heads become the initial trajectory seeds:

$$\begin{aligned} s(q, p) &= \alpha\, \cos\big(e(q), e(p)\big) + (1 - \alpha)\, \mathrm{BM25}(q, p), \qquad p \in \mathcal{M}, \\ \bar{s}(q, C) &= \max_{p' \rightsquigarrow C} s(q, p'), \qquad C \in \mathcal{H}, \\ \Pi_1 &= \mathrm{Top}_{k_0}\big\{(C) : C \in \mathcal{H}\big\} \quad \text{by } \bar{s}(q, C). \end{aligned} \tag{8}$$

Here $e(\cdot)$ denotes a unit-normalized text embedding, and α balances dense and lexical relevance. Because $\rightsquigarrow$ is reflexive, C also contributes its own score. This pooling gives the intra-relation two roles: an earlier expression can retrieve the current form of a fact, while all versions share one seed slot instead of competing with one another. Their accumulated source turns remain attached to C, so temporal changes and conflicting states are still available to the reader. Consequently, a query may match an earlier wording without redirecting retrieval to an obsolete state. Retaining the matched historical member also records why the current head was activated, allowing later trajectory expansion to carry both the latest state and the evidence through which it was recovered.

Inter-relation Links. While intra-relations keep different versions of the same fact aligned, inter-relations connect distinct facts whose relevance emerges only after a seed has been retrieved. Each current head C carries a resolved entity set $\mathcal{E}(C)$. Exact normalized forms share an identifier, while highly similar aliases are linked during indexing and added to the corresponding sets without transitive closure. Ambiguous mentions are resolved from local dialogue context during proposition formation, leaving query-time expansion to use only these precomputed identifiers. Given a trajectory $\pi \in \Pi_{d-1}$, TraceMem considers a current head $C \in \mathcal{H}$ as its next step whenever it shares an identifier with a head already in π. We express this extension point-wise as:

$$\begin{aligned} \pi' &= \begin{cases} (\pi, C), & C \notin \pi,\ \exists C_r \in \pi : \mathcal{E}(C_r) \cap \mathcal{E}(C) \neq \varnothing, \\ \varnothing, & \text{otherwise}, \end{cases} \\ \Pi_d &= \mathrm{Top}_B\big\{\pi' : \pi \in \Pi_{d-1},\ C \in \mathcal{H},\ \pi' \neq \varnothing\big\} \quad \text{by } s(q, \pi'), \qquad 2 \leq d \leq D. \end{aligned} \tag{9}$$

Here (π, C) appends C to π. At each depth, candidates are pre-ranked by their mean member score and capped at κ before batched trajectory scoring; candidates containing the same heads are then

deduplicated by retaining the highest-scoring ordering. The best B complete trajectories survive, up to depth D. If no valid extension remains, retrieval stops and $\Pi^\star$ is the last non-empty beam. Each trajectory is scored in the same way as a seed proposition. The shared entity is used only to expose the next candidate: the candidate is retained according to the relevance of the concatenated trajectory as a whole. Consequently, supporting propositions may be weakly related to q, or to one another, while still contributing necessary evidence. Appendix E reports sensitivity to each retrieval hyperparameter.

Proposition 2 (Trajectory connectivity and lineage preservation)**.** *Every $\pi = (C_1, \ldots, C_d) \in \Pi_d$ consists of distinct current heads, and each C_j for $j \geq 2$ shares at least one entity with a head in its prefix. The earlier versions and source turns of every C_j remain accessible through $p' \rightsquigarrow C_j$. Thus, every trajectory is prefix-connected, lineage-consistent, and provenance-preserving, even when its propositions are not semantically similar.*

As a direct consequence, an evidence chain $F_h = (f_1, \ldots, f_h)$ with $(f_1) \in \Pi_1$ is recoverable up to depth D when its subsequent heads satisfy the bridge condition in Eq. (9), provided that none of its correct prefixes is pruned from the beam. This recovery requires no query-time LLM call.

Reader Input. TraceMem maps $\Pi^\star$ to the top-K source turns cited by its heads, prioritizing each turn by the highest-scoring trajectory that contains it and using its own relevance to break ties, before invoking the reader once:

$$\mathcal{T}_q = \Gamma(\Pi^\star), \qquad \hat{y} = \mathbb{G}(q, \Pi^\star, \mathcal{T}_q)\,. \tag{10}$$

Algorithms 1 and 2 in Appendix A provide the complete indexing and retrieval procedures.

4 Experiments

4.1 Setup

Baselines. We compare with eight representative long-term memory systems: MemGAS (Xu et al., 2026b), MemORAI (Van et al., 2026), A-Mem (Xu et al., 2025), Mem0 (Chhikara et al., 2025), MemoryOS (Kang et al., 2025), SimpleMem (Liu et al., 2026), APEX-MEM (Banerjee et al., 2026), and LightMem (Fang et al., 2026). We reproduce each baseline from its official GitHub repository, preserving the hyperparameter configurations reported in their publications.

Benchmarks and Metrics. We evaluate on LoCoMo-10 (Maharana et al., 2024) and the substantially longer LongMemEval-S (Wu et al., 2025). The former covers multi-hop, temporal, open-domain, single-hop, and adversarial questions, whereas the latter stresses multi-session, single-session, temporal-reasoning, and knowledge-update queries over longer histories. For both benchmarks, we report LLM-judge accuracy by question type and overall as mean $\pm$ standard deviation. Appendix C reports the benchmark distributions, and Appendix D provides additional answer metrics.

Table 1: LoCoMo-10 Judge accuracy (%; ↑ better; mean $\pm$ std). **Bold**/underline: best/second best. Mul.: Multi-hop; Tmp.: Temporal; Ope.: Open-domain; Sgl.: Single-hop; Adv.: Adversarial.

Method	Mul.↑	Tmp.↑	Ope.↑	Sgl.↑	Adv.↑	Overall↑
Mem0	55.48 $\pm$ 0.67	61.68 $\pm$ 0.29	41.66 $\pm$ 1.12	73.72 $\pm$ 0.10	76.49 $\pm$ 0.36	68.31 $\pm$ 0.21
MemoryOS	57.79 $\pm$ 0.44	70.23 $\pm$ 0.53	68.61 $\pm$ 0.85	71.90 $\pm$ 0.22	74.53 $\pm$ 0.39	70.06 $\pm$ 0.30
A-Mem	49.50 $\pm$ 0.33	47.40 $\pm$ 0.44	38.50 $\pm$ 0.49	65.30 $\pm$ 0.10	61.60 $\pm$ 0.42	58.08 $\pm$ 0.25
LightMem	45.37 $\pm$ 0.29	51.59 $\pm$ 0.34	41.66 $\pm$ 0.76	62.87 $\pm$ 0.12	70.12 $\pm$ 0.27	59.20 $\pm$ 0.11
MemGAS	31.21 $\pm$ 0.58	33.96 $\pm$ 0.47	42.39 $\pm$ 0.63	61.36 $\pm$ 0.35	79.82 $\pm$ 0.22	55.91 $\pm$ 0.28
SimpleMem	53.00 $\pm$ 0.41	67.00 $\pm$ 0.36	70.50 $\pm$ 0.69	79.50 $\pm$ 0.22	74.00 $\pm$ 0.33	72.05 $\pm$ 0.24
MemORAI	41.87 $\pm$ 0.34	65.49 $\pm$ 0.28	68.25 $\pm$ 0.45	57.12 $\pm$ 0.31	**90.50 $\pm$ 0.17**	64.33 $\pm$ 0.26
APEX-MEM	58.65 $\pm$ 0.37	59.03 $\pm$ 0.42	65.00 $\pm$ 0.51	77.00 $\pm$ 0.26	86.77 $\pm$ 0.31	73.12 $\pm$ 0.23
TraceMem	**59.93 $\pm$ 0.29**	**75.08 $\pm$ 0.21**	**72.83 $\pm$ 0.33**	**80.68 $\pm$ 0.18**	85.76 $\pm$ 0.24	**77.60 $\pm$ 0.19**

Implementation Details. We use Llama-3.3-70B-Instruct (Meta AI, 2024) for memory transformations and answer generation, Qwen3-Embedding-0.6B (Zhang et al., 2025) for retrieval embeddings, and `gpt-4.1-mini` for judging. All local inference is hosted on a single H100 GPU, and Tables 1 and 4 report mean $\pm$ standard deviation over three independent runs. Every method uses the same reader and judge prompts. Because some baselines provide summaries, graph-derived records, or other memory units instead of raw turns, we strive for a fair comparison by measuring the total tokenized context presented to the reader rather than the number of retrieved units. Table 7 in Appendix E reports the complete configuration.[1]

4.2 Main Results

Improved reasoning through memory trajectories. Table 1 shows that TraceMem reaches 77.60% overall accuracy on LoCoMo-10. It exceeds A-Mem by 19.52 points, MemORAI by 13.27, and APEX-MEM by 4.48, corresponding to a 6.1% relative improvement over APEX-MEM. It also leads four of the five categories, improving over the strongest competing score by 1.28 points on multi-hop, 4.85 on temporal, 2.33 on open-domain, and 1.18 on single-hop questions. By connecting supporting propositions before returning their source context, memory trajectories expose evidence that independent relevance scores leave fragmented.

Notable gains persist over longer histories. The same pattern emerges on LongMemEval-S, where relevant evidence is distributed across more sessions and substantially longer conversations. TraceMem reaches 87.38% overall accuracy in Table 4 of Appendix D, exceeding A-Mem by 24.12 points, MemORAI by 8.12, and APEX-MEM by 1.18. Relative to the strongest competing category scores, it gains 1.30 points on multi-session, 1.50 on single-session, 0.62 on temporal, and 1.12 on knowledge-update questions. These gains across every reasoning regime yield the best overall result, showing that memory trajectories remain effective as cross-session dependencies and temporal distance increase.

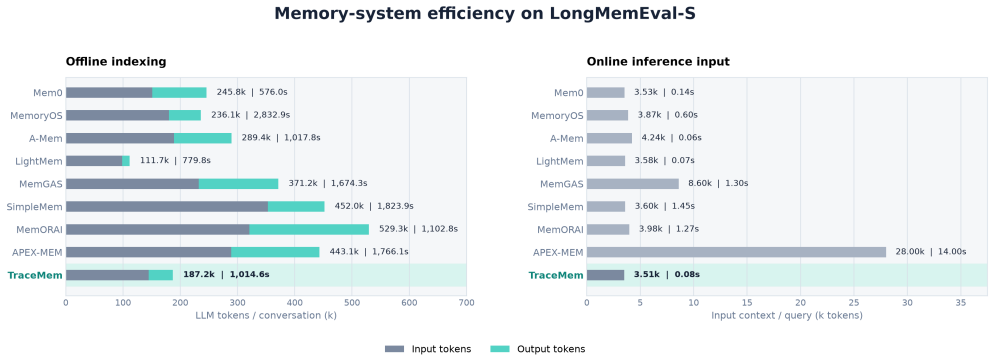

Figure 3: Memory-system efficiency on LongMemEval-S for all main baselines. The left panel separates input and output tokens for offline indexing and reports construction time per conversation. The right panel reports the total input context and retrieval latency per query before answering; QA and judging are excluded.

Near-zero-latency retrieval. TraceMem reaches the strongest accuracy without inserting an online reasoning loop between the question and the reader. Figure 3 shows that, relative to APEX-MEM, it contracts the online input from 28.00k to 3.51k tokens and retrieval latency from 14.00s to 0.08s, yielding an 8.0$\times$ smaller context and a 175$\times$ speedup. Its 187.2k-token offline footprint is also the smallest, 20.7% below MemoryOS. Even Mem0, with an almost identical 3.53k-token context, is 1.75$\times$ slower and trails TraceMem by 9.29 points on LoCoMo-10 and 10.28 points on LongMemEval-S. The resulting trade-off is TraceMem's central selling point: stronger evidence reconstruction with effectively instantaneous retrieval, no complex graph reasoning, and no iterative LLM calls before the reader.

[1] Our code will be made publicly available upon acceptance.

4.3 Ablation Study

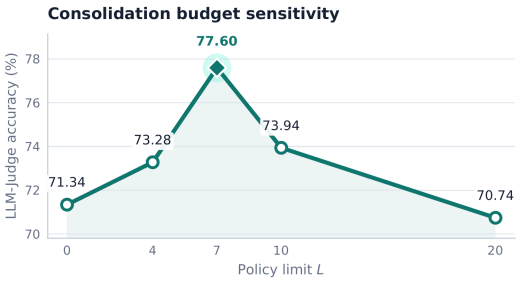

Figure 4: LLM-Judge accuracy across shared consolidation budgets on LoCoMo-10.

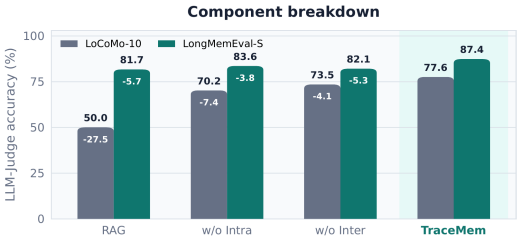

Figure 5: Accuracy contribution of intra- and inter-relations at 20 retrieved turns.

Consolidation Limit. Figure 4 shows that consolidation must connect dispersed evidence without erasing retrievable distinctions. Too little leaves evidence across sessions and topic segments fragmented, preventing complete trajectories; at $L = 0$, accuracy drops by 6.26 points. Too much repeatedly compresses distinct observations into a summary-like head and removes targeted cues; at $L = 20$, accuracy drops by 6.86 points. The peak at $L = 7$ balances connection with retrievable detail.

Component Breakdown. Figure 5 isolates the contribution of each relation. Raw-turn RAG misses distributed evidence and trails TraceMem by $27.55/5.68$ points on LoCoMo-10/LongMemEval-S. Removing intra-relations prevents historical expressions from redirecting to current heads, costing $7.37/3.78$ points; removing inter-relations blocks semantically distant support, costing $4.10/5.28$. Together, both relations yield state-aware trajectories with complementary cross-session evidence.

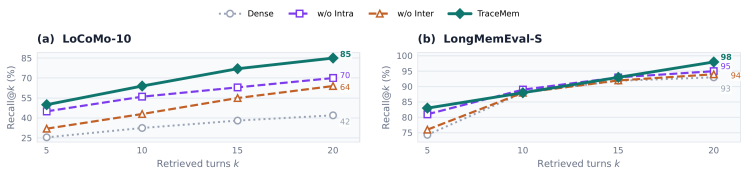

Figure 6: Turn Recall@k across both benchmarks. At $k = 5$, TraceMem already recovers more gold evidence on LoCoMo-10 than dense retrieval with twenty turns.

Trajectory Rescue. Figure 6 complements Figure 5 at the retrieval level. On LoCoMo-10, dense retrieval recovers 42% of gold turns at $k = 20$; intra-relations alone raise this to 64% and inter-relations alone to 70%, while both reach 85%, 21 and 15 points above either alone. On LongMemEval-S, where dense retrieval finds 93%, neither relation alone exceeds 95%, yet their combination reaches 98%. The gain belongs to the memory trajectory: lineage fixes *which* state is current, bridges carry it *across* sessions, and only chaining both reaches what neither exposes alone.

5 Conclusion

TraceMem provides a lightweight conversational memory that reconstructs distributed evidence as memory trajectories over an evolving proposition store. It achieves 77.60% and 87.38% Judge accuracy on LoCoMo-10 and LongMemEval-S while presenting only 3.51k tokens to the reader and requiring 0.08s for retrieval per query. This combination of accuracy and efficiency stems from two complementary relations: (i) intra-relations connect historical expressions to their current heads, preserving state evolution without discarding earlier evidence; and (ii) inter-relations use bridge entities to unite semantically distant but jointly necessary memories, completing cross-session trajectories without complex graph reasoning or iterative LLM retrieval.

AI USE STATEMENT

Generative AI tools were used only to polish the writing. All research ideas, implementations, experiments, analyses, and conclusions were produced and verified by the authors.

REPRODUCIBILITY STATEMENT

Section 4.1 specifies the model stack and evaluation protocol. Appendix D reports additional answer metrics, while Appendix E details the configuration and retrieval sensitivity. The submission includes figure data, deterministic rendering scripts, retrieval outputs, and judge decisions; an anonymized repository will be released with the paper.

REFERENCES

Pratyay Banerjee, Masud Moshtaghi, Shivashankar Subramanian, Amita Misra, and Ankit Chadha. APEX-MEM: Agentic semi-structured memory with temporal reasoning for long-term conversational AI. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16470–16489. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.749. URL https://aclanthology.org/2026.acl-long.749/.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In *ECAI 2025: 28th European Conference on Artificial Intelligence*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pp. 2993–3000. IOS Press, 2025. doi: 10.3233/FAIA251160. URL https://doi.org/10.3233/FAIA251160.

Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Babu Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A. Huerta, and Hao Peng. Context length alone hurts LLM performance despite perfect retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 23281–23298. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.1264. URL https://aclanthology.org/2025.findings-emnlp.1264/.

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://proceedings.iclr.cc/paper_files/paper/2026/hash/a05b72653ec5b473732129829ae04195-Abstract-Conference.html.

Bernal Jiménez 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, 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html.

Nguyen Manh Hieu, Vu Lam Anh, Hung Pham Van, Nam Le Hai, Linh Ngo Van, Nguyen Thi Ngoc Diep, and Thien Huu Nguyen. MaGiX: A multi-granular adaptive graph intelligence framework for enhancing cross-lingual RAG. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 5202–5219. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.279. URL https://aclanthology.org/2025.findings-emnlp.279/.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, and Yafeng Deng. EverMemOS: A self-organizing memory operating system for structured long-horizon reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45836–45853. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.2125. URL https://aclanthology.org/2026.acl-long.2125/.

Dongming Jiang, Yi Li, Guanpeng Li, and Bingzhe Li. MAGMA: A multi-graph based agentic memory architecture for AI agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36848–36865. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1709. URL https://aclanthology.org/2026.acl-long.1709/.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1318. URL https://aclanthology.org/2025.emnlp-main.1318/.

Cai Ke, Yiming Du, Bin Liang, Yifan Xiang, Lin Gui, Zhongyang Li, Baojun Wang, Yue Yu, Hui Wang, Kam-Fai Wong, and Ruifeng Xu. Flexibly utilize memory for long-term conversation via a fragment-then-compose framework. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21119–21136. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1069. URL https://aclanthology.org/2025.emnlp-main.1069/.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. SimpleMem: Efficient lifelong memory for LLM agents. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=oBgLvd5YC6.

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. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Meta AI. Llama 3.3 70B Instruct. Hugging Face model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct.

Mathis Pink, Vy A. Vo, Qinyuan Wu, Jianing Mu, Javier S. Turek, Uri Hasson, Kenneth A. Norman, Sebastian Michelmann, Alexander Huth, and Mariya Toneva. Temporal context reinstatement drives episodic-like order memory in long-context language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=sycSMgogxM.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8a2acd174940dbca361a6398a4f9df91-Abstract-Conference.html.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Rajan Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.413. URL https://aclanthology.org/2025.acl-long.413/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.557. URL https://aclanthology.org/2023.acl-long.557/.

Hung Pham Van, Nguyen Manh Hieu, Khang Pham Tran Tuan, Nam Le Hai, Linh Ngo Van, Nguyen Thi Ngoc Diep, and Trung Le. MemORAI: Memory organization and retrieval via adaptive graph intelligence for LLM conversational agents. In *Findings of the Association for*

Computational Linguistics: ACL 2026, pp. 28235–28253. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1408. URL https://aclanthology.org/2026.findings-acl.1408/.

Kaixiang Wang, Yidan Lin, Zihan Wang, Bunyod Suvonov, Zhaojiacheng Zhou, Yuxiang Zheng, Jiaxi Cao, Zhiheng Dong, Chentao Wu, Jiong Lou, and Jie Li. E-mem: Multi-agent based episodic context reconstruction for LLM agent memory. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=FAjA0snAYq.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-3259. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ebd82705f44793b6f9ade5a669d0f0bf-Abstract-Conference.html.

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://proceedings.iclr.cc/paper_files/paper/2025/hash/d813d324dbf0598bbdc9c8e79740ed01-Abstract-Conference.html.

Buqiang Xu, Yijun Chen, Jizhan Fang, Ruobin Zhong, Yunzhi Yao, Yuqi Zhu, Lun Du, and Shumin Deng. StructMem: Structured memory for long-horizon behavior in LLMs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 122–146. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-short.12. URL https://aclanthology.org/2026.acl-short.12/.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In *International Conference on Learning Representations (ICLR)*, 2026b. URL https://openreview.net/forum?id=i2yIvZARnG.

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*, volume 38, 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models, 2025. URL https://huggingface.co/Qwen/Qwen3-Embedding-0.6B.

A THEORETICAL PROOFS

Proof of Proposition 1. Let $\mathcal{L}$ be the leaf set of the full b-ary bridge tree in Proposition 1, so $|\mathcal{L}| = b^{D-1}$. Each leaf $\lambda \in \mathcal{L}$ determines a unique path $P_\lambda = (C_1^\lambda, \ldots, C_D^\lambda)$. The question q identifies the common root but contains no information that distinguishes the branches below it. Instead, each non-leaf C_d^λ reveals the bridge that identifies only its correct successor C_{d+1}^λ.

We prove by induction that after depth d, active retrieval has retained the correct prefix $(C_1^\lambda, \ldots, C_d^\lambda)$. At $d = 1$, the question identifies the root. Assume the claim holds at some $d < D$. The last recovered head exposes the unique bridge to C_{d+1}^λ, so the next evidence-conditioned step generates and selects that child. The retained trajectory at depth $d + 1$ is therefore the correct prefix extended by one head. Induction reaches P_λ at depth D, and hence the answer-bearing leaf, while retaining exactly D heads. The distinguishing information required at depth $d + 1$ need not be present in q because it is supplied by the prefix already recovered.

Now let a passive retriever return a set $A(q)$ containing at most k heads. Since all choices are fixed by q and q is identical across the possible leaves, $A(q)$ cannot depend on the realized answer leaf λ. Because the answer is stored only at λ, success requires $\lambda \in A(q) \cap \mathcal{L}$. Under a uniform answer leaf,

$$\Pr_{\lambda \sim \mathrm{Unif}(\mathcal{L})}[\text{success}] = \frac{|A(q) \cap \mathcal{L}|}{|\mathcal{L}|} \leq \min\left\{1, \frac{k}{b^{D-1}}\right\}.$$

The same bound holds for a randomized passive retriever whose internal randomness is independent of the realized leaf: conditioning on that randomness yields a fixed set $A(q)$, and averaging preserves the bound. To guarantee recovery for every possible leaf, $A(q)$ must contain all of $\mathcal{L}$, which requires $k \geq b^{D-1}$. Thus the separation is not caused by a stronger reader or a larger retained context: active retrieval converts each recovered proposition into the cue for the next decision, whereas passive retrieval must reserve capacity for branches that the original question cannot distinguish. □

Proof of Proposition 2. For a trajectory $\pi = (C_1, \ldots, C_d)$, consider the following four invariants:

(i) *current-head membership*: $C_j \in \mathcal{H}$ for every j;

(ii) *distinctness*: $C_j \neq C_r$ whenever $j \neq r$;

(iii) *prefix connectivity*: for every $j \geq 2$, there exists $r < j$ such that $\mathcal{E}(C_r) \cap \mathcal{E}(C_j) \neq \varnothing$;

(iv) *provenance preservation*: every historical proposition p' satisfying $p' \rightsquigarrow C_j$, together with its source turns, remains accessible through C_j.

We show by induction that every member of Π_d satisfies all four. At depth one, each $\pi \in \Pi_1$ is a singleton (C_1) selected from $\mathcal{H}$. Membership therefore holds directly; distinctness and prefix connectivity are vacuous. Equation (6) retains every predecessor in persistent memory and records its lineage to the active head, so the singleton also preserves the provenance of C_1.

Assume that the invariants hold for every trajectory in Π_{d-1}, and consider an admissible extension $\pi' = (\pi, C)$ generated by Equation (9). The candidate is drawn from $\mathcal{H}$, which establishes current-head membership for the new element. The condition $C \notin \pi$ preserves pairwise distinctness. The same equation requires a head $C_r \in \pi$ with a shared resolved entity, so C attaches to the existing prefix rather than forming a disconnected component. All earlier heads retain their inductive connectivity, and the new head has an explicit link to one of them; hence the complete extension is prefix-connected.

It remains to establish lineage consistency and provenance. Consolidation changes the active view $\mathcal{H}$ but never removes propositions from $\mathcal{M}$. Under either add or sup, the selected predecessor heads and the incoming proposition become historical members whose lineages terminate at the new current head. Their source turns and entities are inherited by that head. Under noop, the proposition is itself a current head with its own source turns. Consequently, for every candidate C, following $p' \rightsquigarrow C$ recovers all historical members and their grounded evidence. Appending C therefore adds provenance without invalidating that of any earlier trajectory member.

Candidate deduplication retains the highest-scoring ordering for each identical head set, while the cap κ and top-B selection only discard complete candidates; none of these operations edits a surviving candidate. Every member of Π_d is thus an admissible extension and retains the four invariants. This completes the structural induction for all $1 \leq d \leq D$.

We finally prove the stated recovery consequence. Let $F_h = (f_1, \ldots, f_h)$ be an evidence chain with $h \leq D$, $(f_1) \in \Pi_1$, and, for every $d \geq 2$, let f_d share a resolved entity with some head in $F_{d-1} = (f_1, \ldots, f_{d-1})$. Suppose each correct prefix remains among the top B candidates after deduplication and candidate capping.

The base prefix F_1 is present by assumption. If F_{d-1} survives, the bridge condition generates its extension by f_d, and the survival assumption retains that extension in Π_d. Induction reaches a trajectory containing f_h. Head-set deduplication does not alter this conclusion because it retains the highest-scoring ordering over the same evidence heads, and any later bridge may connect to any member of the prefix. Hence the answer-bearing head is recovered, while the structural argument above guarantees that the retained trajectory remains current-head-only, distinct, connected, and grounded in its lineage source turns. □

Corollary 3 (Recovery of query-implicit evidence)**.** *In the bridge-tree construction of Proposition 1, suppose the root is selected as a seed and the correct prefix survives at every beam depth. TraceMem recovers the answer-bearing leaf even when that leaf has arbitrarily low direct relevance to q. Moreover, the recovered trajectory satisfies all structural and provenance invariants of Proposition 2.*

Proof. Proposition 1 ensures that each retrieved prefix exposes the unique bridge needed to generate the next correct head. The recovery induction in Proposition 2 therefore carries the seed to the answer leaf, while its structural induction applies to every surviving prefix. Direct query–leaf similarity is never required after seed selection: it affects the joint trajectory score, but candidate exposure follows the bridges revealed by already retrieved evidence. Thus trajectory expansion can recover evidence that a query-only ranking does not expose, without sacrificing current-state consistency or source-turn grounding. □

A.1 End-to-End Pipeline Pseudocode

Algorithms 1 and 2 make explicit the complete write- and query-time procedures. The implementation stores lineage successors, normalized entity identifiers, pairwise entity-equivalence links, and an entity-to-head inverted lookup alongside the flat proposition records; trajectory retrieval consults these indexes without invoking a global graph algorithm.

Entity-resolution index. Resolution is scoped to one conversation. Its candidate pool contains the distinct normalized entity forms attached to current heads: exact forms share one identifier, while the remaining forms are embedded with Qwen3-Embedding-0.6B and linked pairwise to their five nearest candidates only when cosine similarity is at least 0.8. These links are not closed transitively. At retrieval, $\mathcal{E}(C)$ contains both the head's exact identifiers and their direct equivalents, allowing aliases to expose one another without collapsing an entire similarity chain into one identity. Pronouns and locally ambiguous references are resolved upstream from the surrounding dialogue when the proposition is formed. A form occurring in at least five heads and more than 20% of the active heads is retained as proposition metadata but excluded from bridge expansion, preventing frequent names or generic entities from connecting most of the memory.

Algorithm 1 Incremental Memory Encoding

Require: Chronological sessions $\{\mathcal{S}_i, d_i\}_{i=1}^{n}$; $\mathcal{F}_\theta$; text encoder e; k_c, η, L; entity threshold $\rho = 0.8$; neighbor cap $k_e = 5$
Ensure: Persistent memory $\mathcal{M}$, current heads $\mathcal{H}$, lineage successors, entity lookup $I_\mathcal{E}$

$\mathcal{M} \leftarrow \varnothing$; $\mathcal{H} \leftarrow \varnothing$
for $i = 1, \ldots, n$ **do**
  $\mathcal{Z}_i \leftarrow \mathcal{F}_\theta(\mathcal{S}_i)$; validate that spans are contiguous, disjoint, ordered, and cover $\mathcal{S}_i$
  for $z_{i,j} \in \mathcal{Z}_i$ **do**
    $(\mathcal{P}_{i,j}, \{\mathcal{E}(p)\}) \leftarrow \mathcal{F}_\theta(z_{i,j}; d_i)$; attach layer, date, and source-turn provenance
    for $p \in \mathcal{P}_{i,j}$ in dialogue order **do**
      Canonicalize every entity surface form in $\mathcal{E}(p)$; compute and cache $e(p)$
      $\mathcal{H}_p \leftarrow \mathrm{Top}_{k_c}\{C \in \mathcal{H} : \max_{p' \rightsquigarrow C} \cos(e(p), e(p')) > \eta\}$
      Set $a \leftarrow$ noop if $\mathcal{H}_p = \varnothing$; otherwise obtain $(a, \mathcal{H}'_p, C_p) \leftarrow \mathcal{F}_\theta(p, \mathcal{H}_p)$
      Apply Eq. (6); preserve predecessor propositions, lineage successors, unioned entities, and source turns; enforce limit L
    end for
  end for
end for
Assign one identifier to each exact normalized entity form and build its current-head postings
Embed all unique entity names with Qwen3-Embedding-0.6B
for each non-high-frequency entity ϵ **do**
  Retain up to k_e pairwise neighbors ϵ' with $\cos(e(\epsilon), e(\epsilon')) \geq \rho$; do not compute transitive closure
end for
Expand each $\mathcal{E}(C)$ with its retained direct equivalents and build the current-head lookup $I_\mathcal{E}$
return $\mathcal{M}, \mathcal{H}$, lineage successors, $I_\mathcal{E}$, and cached proposition/entity/turn embeddings

Algorithm 2 Memory Trajectory Retrieval

Require: Query q; $\mathcal{M}, \mathcal{H}, I_{\mathcal{E}}$; k_0, B, D, κ, K; score s from Eq. (8)
Ensure: Trajectories $\Pi^\star$, grounded turns $\mathcal{T}_q$, answer $\hat{y}$

Embed q once and compute $s(q,p)$ for every searchable proposition $p \in \mathcal{M}$
Max-pool each lineage: $\bar{s}(q,C) \leftarrow \max_{p' \rightsquigarrow C} s(q,p')$; retain the matched historical member
$\Pi_1 \leftarrow \mathrm{Top}_{k_0}\{(C) : C \in \mathcal{H}\}$ by $\bar{s}(q,C)$; set $\Pi^\star \leftarrow \Pi_1$
for $d = 2, \ldots, D$ **do**
    Initialize candidate map $A \leftarrow \varnothing$
    for $\pi \in \Pi_{d-1}$ **do**
        Gather the exact and pairwise-equivalent entity identifiers of every head in π
        for $C \in \bigcup_{\epsilon \in \bigcup_{C' \in \pi} \mathcal{E}(C')} I_{\mathcal{E}}[\epsilon]$ with $C \notin \pi$ **do**
            Form $\pi' \leftarrow (\pi, C)$ and pre-rank it by mean member score
        end for
    end for
    Keep at most the top κ pre-ranked candidates; **break** if none remain
    Concatenate each candidate's proposition texts in discovery order and embed all candidates in one batch
    Compute dense–BM25 trajectory scores; for each unordered head set, retain its highest-scoring ordering
    Set $\Pi_d \leftarrow \mathrm{Top}_B(A)$; $\Pi^\star \leftarrow \Pi_d$
end for
Deduplicate heads across $\Pi^\star$ while preserving trajectory and within-trajectory order
Map retained heads to their lineage provenance; score each turn by its best containing trajectory, breaking ties by turn relevance
$\mathcal{T}_q \leftarrow \mathrm{Top}_K$ distinct source turns and $\hat{y} \leftarrow \mathbb{G}(q, \Pi^\star, \mathcal{T}_q)$
return $\Pi^\star, \mathcal{T}_q, \hat{y}$

B Prompt Templates

The following are the official prompt templates used by TraceMem. Bracketed fields are replaced by the corresponding dialogue, proposition, evidence, or answer content at runtime.

Proposition Formulation

```
Transform the dialogue unit into compact, self-contained propositions.

1. Remove filler and redundant acknowledgements.
2. Resolve pronouns, short answers, and implicit references to explicit entities using
the preceding context, but never cite that context as provenance.
3. Normalize relative dates against {session_date} into ISO-8601 dates when possible.
4. Preserve the event, participants, place, reason, quantities, and other details
needed to interpret each proposition independently. Do not split one event into
redundant fragments or merge distinct events.
5. Assign exactly one layer: episodic for a dated event, action, plan, decision, or
state transition; semantic for a relatively stable attribute, preference, relation,
habit, or fact.
6. Attach explicit entities, an event timestamp or null, and at least one supporting
source-turn index.
7. Treat image captions as descriptions of shared images, not as speaker utterances.

Return only:
{"propositions": [{"layer": "episodic|semantic", "content": "...", "entities": ["..."],
"timestamp": "YYYY-MM-DD|null", "source_turns": [0]}]}

PRECEDING CONTEXT: {context}
DIALOGUE UNIT: {segment}
```

Proposition Consolidation

```
Given one NEW proposition and its most similar CURRENT heads, choose exactly one action.
```

Proposition Consolidation (continued)

```
NOOP: the proposition describes a different person, event, object, or time, or the
relation is uncertain. Keep it as a separate head.
ADD: it contributes complementary detail about the same event, object, plan, or
attribute. Produce one faithful canonical statement containing every supported entity,
date, number, reason, and description.
SUP: it changes, corrects, contradicts, or replaces an earlier value or state. Produce
a current canonical statement while preserving the dated evolution in its lineage.

Never merge distinct occasions merely because they share a topic. Prefer NOOP when a
faithful consolidation would drop detail, exceed two sentences, or require unsupported
inference. Use only the supplied candidate ids and return only:
{"action": "NOOP|ADD|SUP", "candidate_ids": ["..."], "canonical_text": "...|null",
"reason": "..."}

NEW: {new_proposition}
CURRENT HEADS: {candidate_heads}
```

Reader QA

```
Answer the question using the supplied raw dialogue turns. Each turn contains its date,
speaker, and utterance. Read all turns before deciding, combine evidence distributed
across them and resolve relative dates against the date attached to the relevant turn.
For lists and counts questions, collect all supported items and deduplicate them.

Answer exactly "Not answerable." only when the question assigns an event or attribute
to the wrong person, or when the evidence contains contradictory information regarding
the requested topic. Missing repetition of a qualifier is not sufficient reason to
abstain. Return one or two concise sentences containing the complete answer, with no
label, markdown, or explanation.

QUESTION: {question}
RAW DIALOGUE TURNS (date | speaker | utterance):
{turns}
```

Judge

```
SYSTEM:
Judge whether the model response is factually equivalent to the reference answer.

USER:
Return [[yes]] only if the response contains or is equivalent to the complete reference
answer; return [[no]] if it is incorrect or incomplete.

QUESTION:
{question}

REFERENCE ANSWER:
{ground_truth}

MODEL RESPONSE:
{prediction}

Output [[yes]] or [[no]] only.
```

C BENCHMARK DETAILS

LoCoMo-10. LoCoMo evaluates whether an assistant can recover information from conversations extending across many sessions (Maharana et al., 2024). Its ten publicly released conversations contain recurring people, events, images, plans, and user states, so relevant evidence may be separated by long stretches of unrelated dialogue. The QA benchmark distinguishes direct single-hop retrieval, multi-hop aggregation, temporal reasoning, open-domain inference, and adversarial questions whose premise is unsupported or attached to the wrong speaker. The evaluation file used in our experiments contains 1,982 questions, whose distribution is re-

ported in Table 2. The imbalance across categories also motivates reporting per-type results rather than relying on Overall accuracy alone.

Table 2: LoCoMo-10 evaluation statistics by question type.

Type	Single-hop	Multi-hop	Temporal	Open-domain	Adversarial	Total
Questions	841	282	321	92	446	1,982

LongMemEval-S. LongMemEval-S contains 500 independently constructed questions embedded in substantially longer conversational histories (Wu et al., 2025). It tests extraction from a single session, aggregation across sessions, temporal reasoning, knowledge updates, and abstention when supporting evidence is absent. Its single-session questions cover user facts, assistant-provided facts, and preferences; we aggregate all three subtypes into Sgl., matching the main results. Multi-session and temporal questions form more than half of the benchmark, making it particularly suitable for testing trajectory construction over dispersed evidence. Table 3 gives the resulting question distribution.

Table 3: Official LongMemEval-S statistics by question type.

Type	Multi-session	Temporal	Knowledge-update	Single-session	Total
Questions	133	133	78	156	500

D LongMemEval-S Results and Additional Answer Metrics

Table 4 reports the full LongMemEval-S Judge results referenced in Section 4.2. Tables 5 and 6 then provide token-level F1 and unigram BLEU (BLEU-1) between each prediction and its reference answer. These lexical metrics complement the semantic judge by measuring answer overlap. On LoCoMo-10, TraceMem achieves the highest Overall F1 and BLEU-1, exceeding the closest baseline by 3.21 and 3.26 points, respectively, and leading four of the five question categories. The same pattern holds on LongMemEval-S, where TraceMem leads Overall F1 and BLEU-1 by 0.91 and 0.90 points and ranks first in every question category. The gains therefore persist under reference-based lexical metrics, showing that the main results are not an artifact of the LLM Judge alone.

Table 4: LongMemEval-S Judge accuracy (%; ↑ indicates higher is better; mean ± std). **Bold** is best and underlined is second best. Mul.: Multi-session; Sgl.: Single-session; Tmp.: temporal-reasoning; Kno.: knowledge-update.

Method	Mul.↑	Sgl.↑	Tmp.↑	Kno.↑	Overall↑
Mem0	80.21 ± 0.44	72.55 ± 0.63	76.91 ± 0.51	81.24 ± 0.21	77.10 ± 0.28
MemoryOS	77.73 ± 0.36	84.25 ± 0.32	78.84 ± 0.47	85.88 ± 0.26	81.33 ± 0.24
A-Mem	49.93 ± 0.81	86.32 ± 0.21	48.42 ± 0.77	65.17 ± 0.56	63.26 ± 0.45
LightMem	71.74 ± 0.29	63.75 ± 0.24	67.18 ± 0.34	83.12 ± 0.27	69.81 ± 0.21
MemGAS	51.88 ± 0.62	75.64 ± 0.43	48.12 ± 0.69	56.41 ± 0.57	59.00 ± 0.41
SimpleMem	84.25 ± 0.30	86.10 ± 0.24	77.44 ± 0.42	79.85 ± 0.35	82.33 ± 0.20
MemORAI	77.28 ± 0.31	81.34 ± 0.27	76.84 ± 0.35	82.61 ± 0.29	79.26 ± 0.22
APEX-MEM	86.70 ± 0.27	89.50 ± 0.22	82.10 ± 0.34	85.74 ± 0.25	86.20 ± 0.19
TraceMem	**88.00 ± 0.18**	**91.00 ± 0.14**	**82.72 ± 0.25**	**87.03 ± 0.20**	**87.38 ± 0.16**

Table 5: LoCoMo-10 auxiliary answer quality. Each question type is reported with separate F1↑ and BLEU-1↑ columns. **Bold** indicates the best result and underlining the second best. Mul.: Multi-hop; Tmp.: Temporal; Ope.: Open-domain; Sgl.: Single-hop; Adv.: Adversarial.

Method	Mul.		Tmp.		Ope.		Sgl.		Adv.		Overall	
	F1↑	BLEU-1↑	F1↑	BLEU-1↑	F1↑	BLEU-1↑	F1↑	BLEU-1↑	F1↑	BLEU-1↑	F1↑	BLEU-1↑
Mem0	37.95	34.45	42.41	38.91	28.00	24.50	51.08	47.58	56.66	53.66	47.99	44.60
MemoryOS	39.61	36.75	48.57	45.07	49.12	43.90	49.77	46.27	55.13	52.13	49.31	45.93
A-Mem	33.64	30.14	32.13	28.63	25.72	22.22	45.02	41.52	45.05	42.05	40.42	37.04
LightMem	32.08	28.91	36.74	33.12	27.95	24.87	45.91	42.36	51.83	48.71	42.96	39.57
MemGAS	20.47	16.97	22.45	18.95	28.52	25.02	42.18	38.68	59.26	56.26	39.11	35.72
SimpleMem	36.16	32.66	46.24	42.74	48.76	45.26	55.24	51.74	54.72	51.72	50.65	47.26
MemORAI	28.15	24.65	45.15	41.65	47.14	43.64	39.13	35.63	**67.59**	**64.59**	45.32	41.93
APEX-MEM	40.23	36.38	40.50	37.00	44.80	41.30	53.44	49.94	64.68	61.68	51.59	48.16
TraceMem	**41.15**	**37.65**	**52.06**	**48.56**	**50.44**	**46.94**	**56.09**	**52.59**	63.89	60.89	**54.80**	**51.42**

Table 6: LongMemEval-S auxiliary answer quality. Each question type is reported with separate F1↑ and BLEU-1↑ columns. **Bold** indicates the best result and underlining the second best. Mul.: Multi-session; Sgl.: aggregated single-session; Tmp.: Temporal reasoning; Kno.: Knowledge update.

Method	Mul.		Sgl.		Tmp.		Kno.		Overall	
	F1↑	BLEU-1↑	F1↑	BLEU-1↑	F1↑	BLEU-1↑	F1↑	BLEU-1↑	F1↑	BLEU-1↑
Mem0	59.46	56.46	53.64	50.64	56.95	53.95	59.83	61.08	57.03	54.70
MemoryOS	57.57	54.57	62.53	59.53	58.42	58.04	63.77	60.77	60.31	58.01
A-Mem	36.45	33.45	66.53	61.10	35.30	32.30	48.03	45.03	47.34	43.58
LightMem	52.84	49.63	47.92	44.88	49.61	46.55	61.44	58.22	51.79	48.67
MemGAS	37.93	34.93	55.99	52.99	35.07	32.07	41.37	38.37	43.34	40.34
SimpleMem	62.53	59.53	63.94	60.94	57.35	54.35	59.19	56.19	61.07	58.07
MemORAI	57.23	54.23	60.32	57.32	56.90	53.90	61.28	58.28	58.74	55.74
APEX-MEM	64.39	61.39	66.49	63.52	60.90	57.90	63.66	60.66	64.00	61.01
TraceMem	**65.38**	**62.38**	**67.66**	**64.66**	**61.37**	**58.37**	**64.62**	**61.62**	**64.91**	**61.91**

E Configuration Setting and Hyperparameter Sensitivity

Table 7 specifies the configuration used throughout the main experiments. It records the transformation and embedding backbones together with the consolidation limit, candidate controls, trajectory-search parameters, and reader context size. This common configuration provides a reproducible reference point for both benchmarks and isolates retrieval behavior from changes to the memory encoder or reader.

Table 7: Detailed hyperparameter configuration for TraceMem.

Module	**Parameter**	**Value / Description**
Memory Encoding	Transformation model	Llama-3.3-70B-Instruct
	Candidate heads (k_c)	10
	Candidate threshold (η)	0.75
	Consolidation limit (L)	7
Trajectory Retrieval	Embedding model	Qwen3-Embedding-0.6B
	Seed heads (k_0)	15
	Beam width (B)	5
	Maximum depth (D)	4
	Dense-score weight (α)	0.75
	Entity-resolution threshold	0.8
	Candidate cap (κ)	600
	Retrieved source turns (K)	20
Answering and Evaluation	Reader	Llama-3.3-70B-Instruct
	Judge	`gpt-4.1-mini`

Tables 8, 9, 10, and 11 report one-factor sweeps for the four parameters that directly control trajectory construction. Beam width B determines how many prefixes survive, maximum depth D bounds trajectory length, α balances semantic and lexical relevance, and k_0 determines how many current heads seed the search. Each panel reports LLM-Judge accuracy while all unvaried settings follow Table 7.

Table 8: Sensitivity to beam width B. Other configurations stayed the same as Table 7.

(a) LoCoMo-10

B	1	3	5	10
J↑	74.69	76.61	77.60	78.43

(b) LongMemEval-S

B	1	3	5	10
J↑	82.34	85.55	87.38	89.82

Table 9: Sensitivity to maximum trajectory depth D. Other configurations stayed the same as Table 7.

(a) LoCoMo-10

D	1	2	3	4
J↑	72.34	73.44	75.97	77.60

(b) LongMemEval-S

D	1	2	3	4
J↑	83.31	84.44	86.19	87.38

Table 10: Sensitivity to hybrid-score weight α. Other configurations stayed the same as Table 7.

(a) LoCoMo-10

α	0	.25	.50	.75	1
J↑	66.13	70.21	74.54	77.60	77.98

(b) LongMemEval-S

α	0	.25	.50	.75	1
J↑	75.13	84.21	85.66	87.38	85.52

Table 11: Sensitivity to seed-head count k_0. Other configurations stayed the same as Table 7.

(a) LoCoMo-10

k_0	5	10	15	20
J↑	73.45	76.69	77.60	78.24

(b) LongMemEval-S

k_0	5	10	15	20
J↑	83.26	86.47	87.38	88.22

F CONFLICT-GROUNDED ABSTENTION

LoCoMo-10 contains 446 adversarial questions. An abstaining response can receive credit for two fundamentally different reasons: the retriever may expose evidence that contradicts the question's premise, or it may retrieve too little information for the reader to attempt an answer. To separate grounded abstention from retrieval failure, after the standard correctness judgment we employ Llama-3.3-70B-Instruct to inspect the retrieved context and classify each response as conflict-supported, evidence-missing, or not an abstention. Conflict J counts a question only when the original judge accepts the response and the verifier identifies explicit evidence supporting the abstention; ΔN denotes the number of originally correct cases removed by this stricter criterion.

Table 12: Conflict-grounded abstention on the 446 adversarial LoCoMo-10 questions. Adv. J is the original Judge accuracy; Conflict J retains only correctly abstained questions supported by explicit conflicting evidence; ΔN is the number of accepted cases removed.

Method	Adv. J↑	Conflict J↑	$\Delta N \downarrow$
Mem0	76.49	65.28	50
MemoryOS	74.53	60.40	63
A-Mem	61.60	52.18	<u>42</u>
LightMem	70.12	58.69	51
MemGAS	79.82	64.80	67
SimpleMem	74.00	61.22	57
MemORAI	**90.50**	72.34	81
APEX-MEM	<u>86.77</u>	**80.72**	**27**
TraceMem	85.76	<u>76.12</u>	43

Under the stricter criterion, TraceMem retains a Conflict J of 76.12%, the second-highest score in Table 12, while only 43 of its originally accepted answers are removed. This preserves 88.8% of its adversarial Judge accuracy and moves TraceMem ahead of MemORAI once unsupported abstentions are excluded, despite MemORAI obtaining the highest original Adv. J. APEX-MEM degrades the least, losing only 6.05 points and 27 accepted cases, plausibly because its iterative agentic query process can explicitly seek counter-evidence before answering.

G CASE STUDY

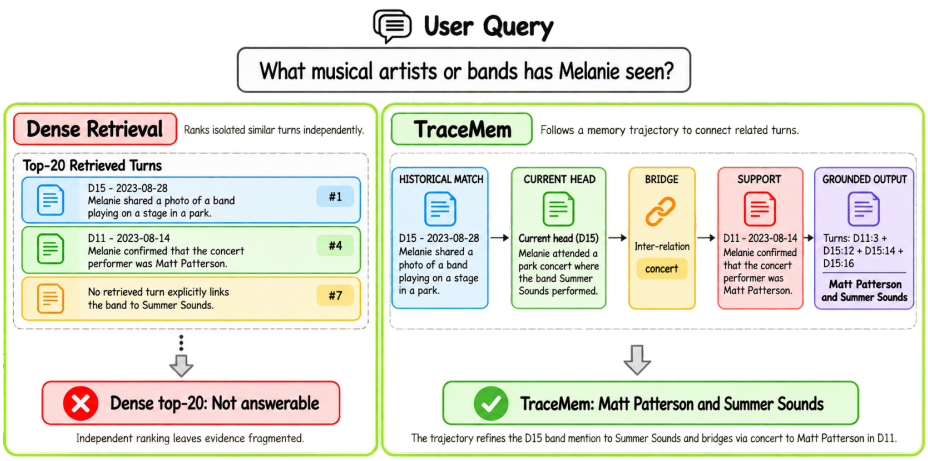

Figure 7: Dense retrieval and TraceMem on `conv-26_q61`. Dense top-20 retrieves relevant but disconnected turns and cannot resolve the unnamed D15 band. TraceMem uses an intra-relation to recover the current *Summer Sounds* head and an inter-relation through *concert* to connect *Matt Patterson* from D11, producing the complete grounded answer.

Figure 7 compares TraceMem with dense top-20 retrieval on the LoCoMo-10 multi-hop question "What musical artists or bands has Melanie seen?" The answer is not stated in one self-contained turn. One session identifies *Matt Patterson*, whereas a later session mentions seeing a band before subsequent turns reveal that the band is *Summer Sounds*. A successful retriever must therefore resolve a locally underspecified reference and then combine it with evidence from another session.

Dense retrieval finds individually relevant fragments but does not reconstruct this dependency. It ranks the D11 turn naming *Matt Patterson* and a D15 turn about seeing an unnamed band, yet the turns that resolve the band's identity do not survive together in the retrieved context. The reader consequently sees two disconnected clues rather than two grounded answer items and returns "Not answerable." This failure is not caused by an absence of relevant turns in the history; it arises because independent turn scores do not represent how one retrieved memory changes the interpretation of another.

TraceMem resolves the ambiguity before constructing the reader context. The historical D15 match follows its intra-relation to the current *Summer Sounds* head, carrying forward the source turns that established the refined identity. From that head, the inter-relation through the shared entity *concert* exposes the D11 proposition naming *Matt Patterson*. Joint trajectory scoring retains the pair because the concatenated propositions answer the question more completely than either proposition alone. Mapping the resulting trajectory back to D11:3 and D15:12/14/16 gives the reader both artists with their original wording, dates, and speakers in a single call.

The example also separates the roles of the two relations. Without the intra-relation, retrieval retains the historical mention but cannot reliably identify the unnamed band; without the inter-relation, the resolved *Summer Sounds* memory remains disconnected from *Matt Patterson*. Their composition turns partial evidence into a complete, provenance-preserving trajectory while leaving the final answer grounded in raw dialogue turns rather than proposition text.