WEAVEMEM: TYPED EXPERIENCE GRAPHS FOR APPLICABILITY REASONING IN LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Persistent language agents rely on evidence distributed across interaction histories that extend far beyond their context windows to answer queries. Existing memory systems retrieve records through semantic similarity or reconstruct context using associative structures. However, similarity retrieval overlooks indirectly useful evidence, while associative reconstruction may recover connected memories without establishing whether their relations are relevant to the query. We formulate this missing capability as *applicability reasoning*: determining whether and how retrieved memories jointly support a query through typed and directional relations that form a compatible and complete evidence chain. To operationalize this capability, we introduce `WeaveMem`, a long-term agent memory system built around a Typed Experience Graph and Applicability Tracing. The Typed Experience Graph grounds typed and directional relations in atomic memories and projects them onto compact experiences with provenance. Applicability Tracing constructs query-specific evidence chains, selects compatible and sufficient traces, and recovers their atomic evidence for answer generation. Across evaluations averaged over five independent runs on LoCoMo, LongMemEval-S, PersonaMem-32K, and BEAM-1M, `WeaveMem` achieves the highest overall score on each benchmark among the evaluated memory systems. Controlled evaluations on relation-dependent queries further show that `WeaveMem` more accurately identifies how retrieved memories relate to one another and jointly constitute complete, valid evidence chains.

1 INTRODUCTION

As language agents evolve from short-lived assistants into persistent systems, they need to reason over interaction histories that extend far beyond the context window. External memory supports this capability by storing past interactions and retrieving a compact context for each query (Packer et al., 2023; Zhong et al., 2024). Early systems primarily adopt *similarity retrieval*, representing memories independently, ranking them by semantic proximity to the query, and supplying the top-ranked items to the answer model (Lewis et al., 2020; Park et al., 2023; Chhikara et al., 2025). This paradigm efficiently recovers directly matched information, but may miss evidence whose utility emerges only through its relation to other memories.

To overcome the limitations, recent systems increasingly adopt *associative reconstruction* to organize memories using semantic links, shared entities, temporal connections, or hierarchical structures, and expand retrieval through these associations (Xu et al., 2025b; Rasmussen et al., 2025; Tang et al., 2026). These methods extend recall beyond direct semantic matches by recovering structurally connected memories. However, they generally optimize associative relevance rather than explicitly validating whether the retrieved relations are applicable to the query and jointly form a directional, compatible, and complete evidence chain. Consequently, the reconstructed evidence may be connected yet not truly applicable to the current query.

We formulate this missing capability as *applicability reasoning*, determining whether and how retrieved memories can support a query by evaluating their relations, directionality, compatibility, and collective completeness. *Applicability reasoning* goes beyond identifying that memories are connected by determining how they are related and what logical implications follow from those

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.

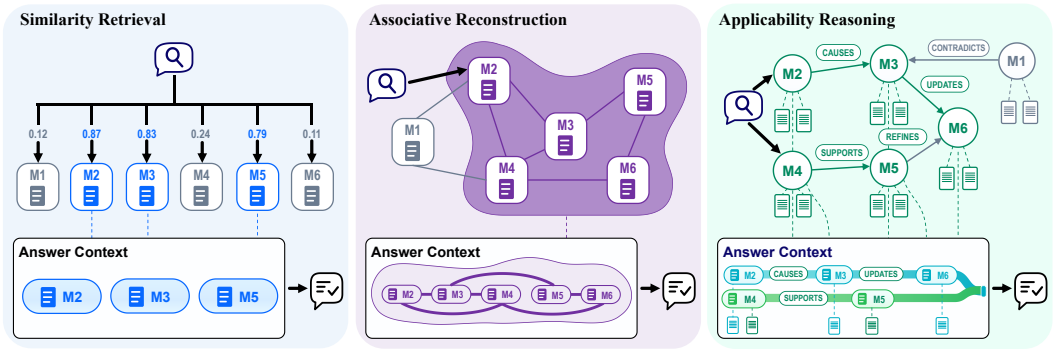


Figure 1: Comparison of three memory-reasoning paradigms. **Left: Similarity retrieval** independently ranks memories by semantic similarity. **Middle: Associative reconstruction** connects related memories through associative links. **Right: Applicability reasoning** selects and composes query-applicable memories into a relation-consistent evidence chain.

relations. It organizes retrieved information into a query-specific evidence chain whose typed and directional relations make explicit the underlying logic connecting the evidence to the answer.

Motivated by this, we propose `WeaveMem`, a long-term agent memory system that supports *applicability reasoning* through two components. (1) **Typed Experience Graph** represents memory content and inter-memory logic at two granularities. It abstracts interaction history into compact experiences, preserves source-linked atomic memories, induces typed and directional relations between atomic memories, and projects these relations onto the experience graph for grounded traversal. (2) **Applicability Tracing** constructs query-specific evidence chains. It retrieves initial experiences, expands query-conditioned traces over typed and directional relations, selects complementary and sufficient traces, and recovers their atomic evidence for answer generation. The graph specifies how memories are related, while tracing determines which relational structure applies to the query.

Extensive experiments across four long-term memory benchmarks demonstrate the strong performance of `WeaveMem`. Averaged over five runs, `WeaveMem` achieves the highest overall benchmark-specific score on every benchmark, reaching 92.29% on LoCoMo (Maharana et al., 2024), 89.00% on LongMemEval-S (Wu et al., 2025), 72.87% on PersonaMem-32K (Jiang et al., 2025), and 74.11% on BEAM-1M (Tavakoli et al., 2026). A controlled evaluation of evidence relations and paths further demonstrates that `WeaveMem` accurately determines how retrieved memories apply to each query, validating its *applicability reasoning* capability. Our contributions are summarized as follows.

- We formulate *applicability reasoning* as a distinct objective for long-term memory systems, requiring agents to determine how memories relate and jointly support a query beyond their semantic relevance or structural connectivity.
- We propose `WeaveMem`, a long-term memory system that builds a typed experience graph to model inter-memory logic and construct query-specific evidence chains.
- We demonstrate that `WeaveMem` consistently outperforms strong memory systems across four long-term memory benchmarks, and validate its *applicability reasoning* capability.
- We provide an anonymized repository with the implementation and configurations to facilitate reproducibility: `https://github.com/WeaveMemory/WeaveMem`.

2 Related Work

Long-Term Agent Memory. Long-term agent memory has evolved from retaining interaction records to selectively maintaining structured representations of experience. Generative Agents (Park et al., 2023) and MemoryBank (Zhong et al., 2024) represent past interactions as textual records or summaries and retrieve them using semantic relevance and temporal or retention signals.

MemGPT (Packer et al., 2023), Mem0 (Chhikara et al., 2025), and REMem (Shu et al., 2026) move beyond passive retrieval by coordinating memory tiers, maintaining consolidated entries, or integrating memory access with reasoning. MemGAS (Xu et al., 2025a) and MemoryOS (Kang et al., 2025) further organize accumulated experience through multi-granularity representations and hierarchical memory lifecycles. This progression improves how experiences are retained, updated, and retrieved. However, evidence required by a query may be distributed across records whose utility depends on how they relate. This has motivated a shift from retrieving memories independently toward organizing and accessing them through relational structures.

Relational Memory Retrieval. Relational memory retrieval represents connections among memories and uses the resulting structures to recover distributed evidence. A-Mem (Xu et al., 2025b) and Graphiti (Rasmussen et al., 2025) construct evolving memory graphs from contextual attributes, shared entities, and temporal relations. HippoRAG 2 (Gutiérrez et al., 2025) combines knowledge-graph associations with graph propagation to recover interconnected evidence. Building on graph-based organization, Mnemis (Tang et al., 2026) and Memora (Xia et al., 2026) represent memory at multiple levels of abstraction to support hierarchical retrieval. HyperMem (Yue et al., 2026) and EverMemOS (Hu et al., 2026) enrich memory structure through higher-order associations or multi-level consolidation. MAGMA (Jiang et al., 2026) separates semantic, temporal, causal, and entity relations into multiple graph views and performs query-adaptive traversal over them. These developments expand relational retrieval from pairwise links to temporal, hierarchical, higher-order, and multi-graph organizations. Nevertheless, existing relations principally support memory organization, candidate expansion, or relevance estimation. Whether the recovered relations jointly constitute a directional, compatible, and sufficient evidence chain remains largely implicit. This leaves the query-conditioned applicability of relational evidence unresolved.

3 METHODOLOGY

3.1 OVERVIEW

Given an interaction history $\mathcal{D}$ and a query q, `WeaveMem` constructs a compact evidence context $\mathcal{C}_q$ for answer generation. The **Typed Experience Graph** organizes $\mathcal{D}$ into experiences $\mathcal{M}$ and source-linked atomic memories $\mathcal{L}$, induces typed directional relations between atomic memories, and projects them onto an experience graph $\mathcal{G}$. **Applicability Tracing** retrieves relevant experiences, constructs and selects query-specific evidence traces $\Pi_q^\star$, and recovers their supporting atomic memories $\mathcal{F}_q$ to form $\mathcal{C}_q$. The answer model then generates $\hat{a}$ from $(q, \mathcal{C}_q)$:

$$\mathcal{D} \to (\mathcal{M}, \mathcal{L}, \mathcal{G}) \xrightarrow{q} \Pi_q^\star \to \mathcal{F}_q \to \mathcal{C}_q \to \hat{a}.$$

3.2 TYPED GRAPH CONSTRUCTION

Hierarchical Experience Representation. `WeaveMem` represents the history at two granularities. It first partitions $\mathcal{D}$ into semantically coherent experiences $\mathcal{M} = \{m_i\}_{i=1}^N$, each containing a topic, summary, retrieval metadata, and pointers to its source turns. These turns are further decomposed into atomic memories

$$\mathcal{L}(m_i) = \mathcal{L}_i = \{\ell_{ij}\}, \qquad \mathcal{L} = \bigcup_{i=1}^{N} \mathcal{L}_i.$$

Each ℓ_{ij} retains its source evidence $\sigma(\ell_{ij})$, parent experience $\rho(\ell_{ij}) = m_i$, and memory type $z(\ell_{ij}) \in \mathcal{Z}$, with $\mathcal{Z} = \{\text{CORE}, \text{EPISODIC}, \text{KNOWLEDGE}\}$.

Grounded Relation Induction. For each atomic memory ℓ_s, `WeaveMem` retrieves candidate memories through lexical, tag-based, and semantic matching. A relation model classifies each candidate pair (ℓ_s, ℓ_t) as unrelated or assigns a relation

$$r \in \mathcal{R} = \{\text{SUPPORTS}, \text{CONTRADICTS}, \text{EXPLAINS}, \text{CAUSES}, \text{PRECEDES}, \text{ENABLES}, \text{UPDATES}\}.$$

An atomic edge is represented as

$$e^{\text{atom}} = (\ell_s, r, \ell_t, d_e, c_e).$$

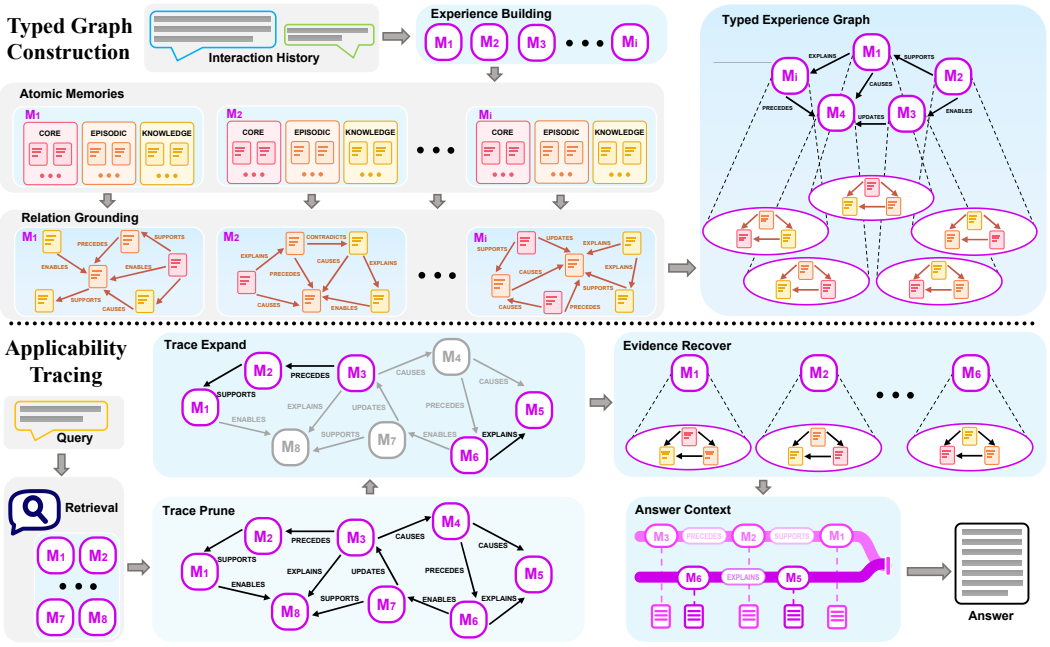


Figure 2: Overview of WeaveMem. **Top: Typed graph construction.** Interaction histories are organized into typed experience graphs. **Bottom: Applicability tracing.** Query-specific traces are expanded, pruned, and converted into answer context.

Here, d_e describes the relation and c_e denotes its confidence. The edge is retained when

$$e^{\text{atom}} \in \mathcal{E}_{\text{atom}} \iff c_e \geq \tau_R \wedge g(e^{\text{atom}}) = 1.$$

The grounding function $g(\cdot)$ verifies source support, subject alignment, temporal consistency, relation direction, and the absence of an equally plausible alternative relation. Entity and temporal checks additionally reject mismatched or reversed relations, producing the atomic graph $\mathcal{G}_{\text{atom}} = (\mathcal{L}, \mathcal{E}_{\text{atom}})$.

Relation Projection. Each atomic edge is projected onto its parent experiences:

$$(\ell_s, r, \ell_t) \mapsto \big(\rho(\ell_s), r, \rho(\ell_t)\big).$$

Intra-experience edges are discarded, while edges sharing the same experience endpoints and relation type are merged into

$$e = (m_s, r, m_t, \mathcal{P}_e).$$

The provenance set $\mathcal{P}_e$ stores their atomic endpoints, descriptions, confidence scores, and source evidence. Asymmetric relations preserve their direction, whereas CONTRADICTS is symmetric. The resulting Typed Experience Graph is $\mathcal{G} = (\mathcal{M}, \mathcal{E})$.

3.3 APPLICABILITY TRACING

Seed Retrieval. Each experience is indexed by its topic, summary, and atomic memories. Given q, WeaveMem independently ranks experiences using dense similarity and BM25 and combines the rankings by Reciprocal Rank Fusion:

$$S_{\text{RRF}}(m) = \sum_{\nu \in \{\text{dense,lexical}\}} \frac{1}{\kappa + \text{rank}_\nu(m)}.$$

The top C_i candidates are reranked, and the highest-scoring K_i experiences form the seed set

$$\mathcal{S}_q = \text{Top}_{K_i}\left(\text{Rerank}\left(q, \text{Top}_{C_i}(S_{\text{RRF}})\right)\right).$$

Each seed initializes a trace and is retained as a zero-hop candidate.

Query-Conditioned Trace Expansion. A query-analysis controller derives a soft relation prior $\mathcal{R}_q \subseteq \mathcal{R}$ from the query. This prior guides traversal without excluding other relations supported by their descriptions. Each seed $m_0 \in \mathcal{S}_q$ initializes $\pi^{(0)} = (m_0)$, and a trace at depth h is

$$\pi^{(h)} = (m_0, e_1, m_1, \ldots, e_h, m_h).$$

At each step, the tracing controller evaluates the current trace and every unvisited frontier edge using its relation type, stored direction, traversal direction, and description. Selected edges extend the trace as

$$\pi^{(h+1)} = \pi^{(h)} \oplus (e_{h+1}, m_{h+1}), \qquad h < H.$$

Edges may be traversed in either direction while retaining their stored direction. The controller may select multiple edges, and expansion stops when the trace is sufficient, no useful edge remains, or depth H is reached. The content of an unvisited node is hidden until its edge is selected.

Trace Selection and Coverage Recovery. For each seed, a trace pruner retains at most P complementary traces and removes redundant or unsupported paths. A sufficiency controller then evaluates whether the retained traces and zero-hop candidates collectively cover the query. If coverage is insufficient, `WeaveMem` reranks a broader pool of C_f experiences, replaces the initial seeds with K_f new seeds, and repeats tracing.

Candidate traces are scored as

$$S(\pi) = w_n S_{\text{node}}(\pi, q) + w_t S_{\text{type}}(\pi, q) + w_d S_{\text{desc}}(\pi, q) + w_c S_{\text{ctrl}}(\pi, q).$$

The four terms measure node relevance, relation-type agreement, relation-description relevance, and the controller assessment, respectively. Traces are greedily selected by score under a budget of B unique experiences. A trace is selected only if all its experiences fit within the remaining budget, yielding the intact trace set $\Pi_q^\star$.

Atomic Evidence Recovery. Let $\mathcal{M}_q^\star$ be the experiences contained in $\Pi_q^\star$. Their atomic memories are globally reranked against the query:

$$\mathcal{F}_q = \text{Top}_{K_{\text{fact}}} \left\{ \ell \in \bigcup_{m \in \mathcal{M}_q^\star} \mathcal{L}(m) \;\middle|\; S_{\text{fact}}(\ell, q) \geq \tau_{\text{fact}} \right\}.$$

The selected atomic memories are grouped by their parent experiences and source evidence to form $\mathcal{C}_q$. The answer model receives only q, the selected experience summaries, and their recovered evidence, and generates

$$\hat{a} = \text{AnswerModel}(q, \mathcal{C}_q).$$

4 Experiments

In this section, we conduct experiments to answer the following research questions.

- **RQ1:** Can `WeaveMem` consistently improve answer quality across long-term memory settings with different history scales and reasoning requirements?
- **RQ2:** Can `WeaveMem` perform applicability reasoning and construct relation-consistent evidence chains beyond semantic relevance or untyped structural connectivity?
- **RQ3:** How efficient is `WeaveMem` in terms of answer-model input token usage?

Datasets and evaluation metrics. We evaluate on four complementary long-term memory benchmarks using their official evaluation sets. LoCoMo (Maharana et al., 2024) covers single-hop, multi-hop, temporal, and open-domain reasoning over multi-session conversations. LongMemEval-S (Wu et al., 2025) emphasizes knowledge updates, cross-session reasoning, temporal reasoning, and session-level information extraction. PersonaMem-32K (Jiang et al., 2025) evaluates personalized responses grounded in evolving user information, while BEAM-1M (Tavakoli et al., 2026) tests diverse memory capabilities over histories approaching one million tokens. Following each benchmark's evaluation protocol, we report overall and category-level benchmark-specific score. For each memory system, we independently generate and evaluate answers over five runs and report the mean and sample standard deviation. Dataset statistics are provided in Appendix B.1. Additional answer-overlap metrics and repeated-run details are provided in Appendix B.2.

Method	LoCoMo				
	Single-hop	Multi-hop	Temporal	Open Domain	Overall
Full Context	94.67	88.44	84.42	76.88	90.29
Mem0	84.71±0.33	68.44±1.76	80.19±1.11	62.71±2.00	79.42±0.69
A-Mem	79.95±0.39	73.33±1.16	77.57±1.19	58.54±2.38	76.91±0.31
Graphiti	62.00±0.51	61.21±0.64	51.53±0.72	63.96±2.28	59.79±0.23
REMem	78.67±0.14	51.91±0.74	82.31±0.26	53.13±1.80	72.94±0.24
Memora	94.77±0.44	**88.79**±0.96	88.91±1.22	72.29±0.93	91.05±0.19
MemGAS	73.10±0.26	44.40±1.26	39.75±1.69	55.21±1.65	59.78±0.43
HyperMem	93.96±0.40	86.45±1.08	86.17±1.70	68.13±0.93	89.35±0.23
EverMemOS	93.32±0.23	86.24±1.10	90.78±0.47	72.08±0.87	90.17±0.32
`WeaveMem` (Ours)	**95.93**±0.46	88.09±1.05	**91.96**±1.02	**73.75**±1.54	**92.29**±0.32
Method	LongMemEval-S				
	KU	MS	SS-Asst	SS-User	Overall
Full Context	81.79	68.12	97.86	94.00	74.80
Mem0	88.72±2.11	66.32±0.82	86.07±1.49	94.86±1.63	81.16±0.30
A-Mem	87.44±1.40	77.59±3.07	95.36±1.60	95.43±0.64	84.16±0.79
Graphiti	66.41±2.11	39.10±1.76	72.50±0.98	83.43±0.78	61.48±0.73
REMem	90.51±0.70	74.74±0.67	78.57±2.53	92.57±0.64	80.60±0.24
Memora	89.49±1.07	**82.11**±1.87	65.36±0.98	98.00±1.63	86.20±0.58
MemGAS	74.36±0.91	53.38±1.19	98.21±0.00	94.86±0.78	72.04±0.43
HyperMem	91.54±1.15	78.35±0.82	91.43±1.49	95.14±0.78	87.20±0.20
EverMemOS	83.59±1.67	77.44±1.76	90.71±1.49	98.29±0.64	86.80±0.20
`WeaveMem` (Ours)	**92.05**±1.40	80.15±1.47	**98.93**±0.98	**99.43**±0.78	**89.00**±0.58
Method	BEAM-1M				
	MS	CR	PF	IE	Overall
Full Context	60.22	42.93	80.43	68.33	51.74
Mem0	66.34±0.88	66.86±0.71	86.63±0.87	72.54±1.13	64.23±0.23
A-Mem	74.54±1.41	54.50±1.67	88.56±1.34	72.53±1.71	65.60±0.37
Graphiti	60.32±0.88	53.50±0.51	80.79±1.30	64.71±0.87	58.54±0.30
REMem	65.71±0.85	70.18±1.11	85.04±1.39	72.41±1.15	64.72±0.40
Memora	72.19±1.37	65.32±1.11	91.44±1.21	78.32±1.36	73.64±0.23
MemGAS	62.17±0.55	45.86±0.51	85.79±0.91	59.28±0.65	55.97±0.21
HyperMem	**78.66**±1.52	61.36±0.39	90.83±1.34	75.78±1.64	70.63±0.73
EverMemOS	74.05±1.13	57.93±0.92	90.30±0.83	77.24±1.31	71.10±0.18
`WeaveMem` (Ours)	76.75±0.46	**73.75**±1.26	**91.60**±0.38	**93.44**±0.94	**74.11**±0.16
Method	PersonaMem-32K				
	Gen.	Rec.	Reasons	Evolution	Overall
Full Context	80.70	72.73	91.92	71.08	66.83
Mem0	71.93±0.00	76.73±1.52	86.06±0.85	67.05±0.32	69.00±0.26
A-Mem	85.26±0.96	78.55±1.99	87.07±0.85	72.37±1.09	70.42±0.42
Graphiti	72.28±0.78	69.82±1.00	79.60±1.81	58.56±0.64	56.57±0.22
REMem	61.05±2.88	74.18±0.81	81.21±1.15	54.68±0.51	62.85±0.66
Memora	74.74±1.57	73.82±2.07	86.67±0.45	73.38±0.72	70.59±0.35
MemGAS	59.30±2.29	58.18±1.82	72.12±0.90	52.81±2.07	58.47±0.44
HyperMem	73.68±1.24	72.36±1.99	**90.71**±0.45	**73.67**±1.66	71.10±0.62
EverMemOS	86.32±0.78	81.82±2.23	85.86±1.24	68.78±0.82	69.17±0.50
`WeaveMem` (Ours)	**87.72**±2.15	**85.45**±1.82	88.08±1.11	70.36±1.07	**72.87**±0.62

Table 1: Mean benchmark-specific scores (0–100) with standard deviations across four benchmarks.

Baselines. We compare `WeaveMem` with eight representative long-term memory systems. Mem0 (Chhikara et al., 2025) uses fact-based memory, while A-Mem (Xu et al., 2025b) dynamically organizes memories through evolving associations. Graphiti (Rasmussen et al., 2025) constructs a temporal knowledge graph, and REMem (Shu et al., 2026) integrates episodic retrieval

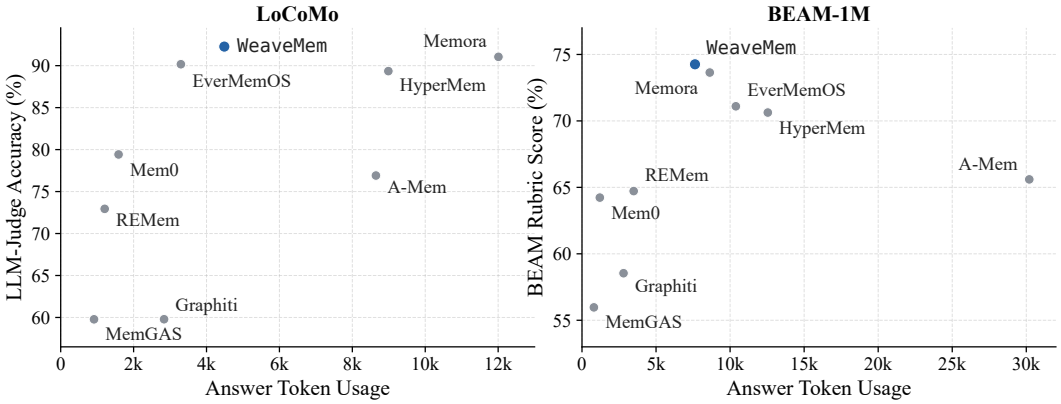


Figure 3: Overall benchmark-specific scores versus average answer-model input tokens on LoCoMo and BEAM-1M.

Method	Provenance Recall ↑	Applicable Precision ↑	Sufficient Context ↑	Answer Acc. ↑
Semantic Retrieval	47.8%	71.5%	51.2%	58.7%
Untyped Graph Traversal	44.9%	59.9%	50.0%	53.6%
`WeaveMem` w/o Applicability	50.2%	59.7%	54.0%	60.7%
Full `WeaveMem`	**54.5%**	**77.6%**	**61.9%**	**65.5%**

Table 2: Query-conditioned evidence applicability on LoCoMo using provenance-level evidence.

with reasoning. MemGAS (Xu et al., 2025a), Memora (Xia et al., 2026), HyperMem (Yue et al., 2026), and EverMemOS (Hu et al., 2026) employ multi-granularity, hierarchical, hypergraph-based, and self-organizing memory structures, respectively. We also include Full Context, which directly supplies the available history subject to the answer model's context limit. Implementation sources, benchmark adaptations, and configurations are detailed in Appendix B.3.

Implementation details. All methods use `GPT-4o-mini` (OpenAI, 2024) with the same answer prompt and benchmark-specific evaluation protocol; outputs are evaluated by `GPT-4.1-mini` (OpenAI, 2025). `WeaveMem` uses `text-embedding-v4` (Alibaba Cloud, 2026) for dense retrieval and `GLM-5.2` (Z.ai, 2026) for graph construction and applicability tracing. Its default configuration uses $K_i = 5$ seeds, $P = 5$ traces per seed, depth $H = 3$, experience budget $B = 10$, relation threshold $\tau_R = 0.95$, and atomic-evidence threshold and budget $(\tau_{\text{fact}}, K_{\text{fact}}) = (0.5, 10)$. Model and decoding configurations, complete hyperparameters, and prompts are provided in Appendices B.4, B.6, and D, respectively.

Main results (RQ1). As shown in Table 1, `WeaveMem` achieves the highest mean overall benchmark-specific overall scores on all four benchmarks, outperforming the strongest memory baseline by 1.24 percentage points on LoCoMo, 1.80 on LongMemEval-S, 1.77 on PersonaMem-32K, and 0.47 on BEAM-1M. At the category level, `WeaveMem` leads in most reported domains, showing particular strengths in conflict resolution, evidence extraction, and personalized response. Its consistent overall performance demonstrates strong generality across diverse history scales and memory requirements. Additional domain-level analyses and complementary evaluation results are provided in Appendix C.4.

Applicability reasoning (RQ2). Applicability reasoning extends relevance retrieval from identifying related memories to selecting query-applicable evidence and composing relation-consistent evidence chains. Semantic retrieval and untyped graph traversal serve as reference baselines, while the controlled comparison between Full `WeaveMem` and `WeaveMem` w/o Applicability uses matched candidate memories and answer-model conditions to isolate the contribution of applicability reason-

Method	Complete Valid Trace ↑	Relation Selection Acc. ↑	Direction Validity ↑	Invalid Edge ↓	Answer Acc. ↑
`WeaveMem` w/o Applicability	86.5%	85.7%	92.2%	49.7%	80.8%
Full `WeaveMem`	**94.0%**	**99.1%**	**99.5%**	**1.0%**	**89.8%**

Table 3: Relation-aware trace recovery on MQuAKE-T with explicit multi-hop fact chains.

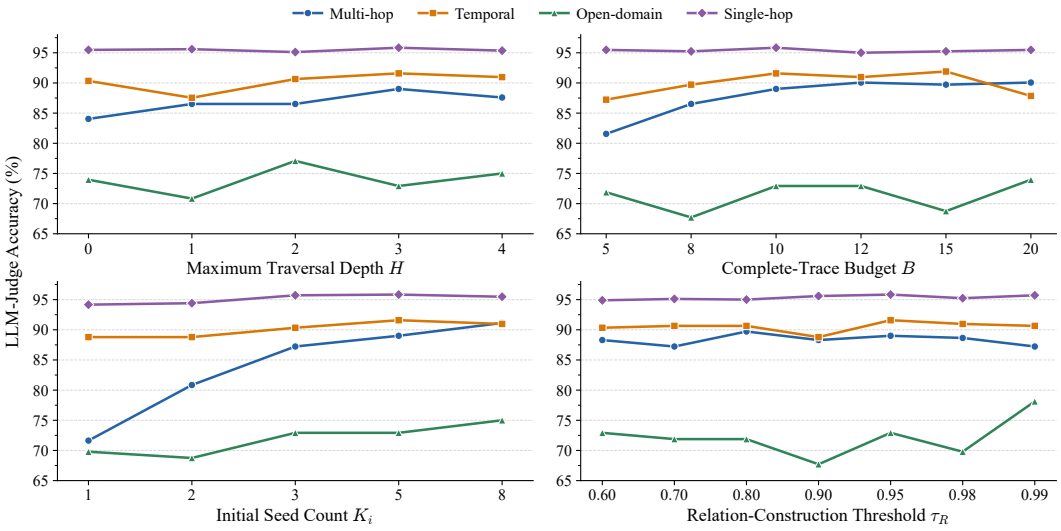


Figure 4: Hyperparameter sensitivity on LoCoMo.

ing. On LoCoMo, Table 2 evaluates evidence selection in natural conversations without assuming a unique gold path. Full `WeaveMem` achieves the best results on all four metrics, reaching 54.5% provenance recall, 77.6% applicable precision, 61.9% sufficient-context rate, and 65.5% answer accuracy. On MQuAKE-TZhong et al. (2023), whose explicit multi-hop fact chains provide gold relation types, directions, and chain order, Table 3 evaluates relational trace construction using complete valid trace rate, relation accuracy, direction accuracy, invalid-edge rate, and answer accuracy; hop-stratified results are provided in the appendix. Full `WeaveMem` improves the complete valid trace rate from 86.5% to 94.0%, relation accuracy from 85.7% to 99.1%, and direction accuracy from 92.2% to 99.5%. It also increases answer accuracy from 80.8% to 89.8% while reducing the invalid-edge rate from 49.7% to 1.0%. Together, these results show that `WeaveMem` filters memories that are merely related but inapplicable and composes applicable evidence into relation-consistent chains, establishing applicability reasoning as a measured capability rather than an inference based solely on QA gains.

Efficiency (RQ3). Figure 3 compares overall accuracy with the average input tokens supplied to the answer model. Among methods with competitive accuracy, `WeaveMem` achieves the highest accuracy on LoCoMo with 4.49K tokens, using 50.1% fewer than HyperMem and 62.6% fewer than Memora. On BEAM-1M, it achieves the highest accuracy with 7.63K tokens, using 11.6% fewer than Memora and 26.6% fewer than EverMemOS. Overall, `WeaveMem` achieves the best performance with a moderate answer context size. Results on LongMemEval-S and PersonaMem-32K are reported in Appendix C.2.

Hyperparameter sensitivity. We analyze sensitivity on LoCoMo by varying one parameter at a time. Figure 4 examines the traversal depth H, experience budget B, initial seed count K_i, and relation threshold τ_R. Across most parameter values and question categories, accuracy changes only moderately, indicating that `WeaveMem` does not depend on precise hyperparameter tuning. The main sensitivity occurs in multi-hop questions when traversal is severely restricted or too few seeds are retrieved; once sufficient graph reach and seed coverage are established, fur-

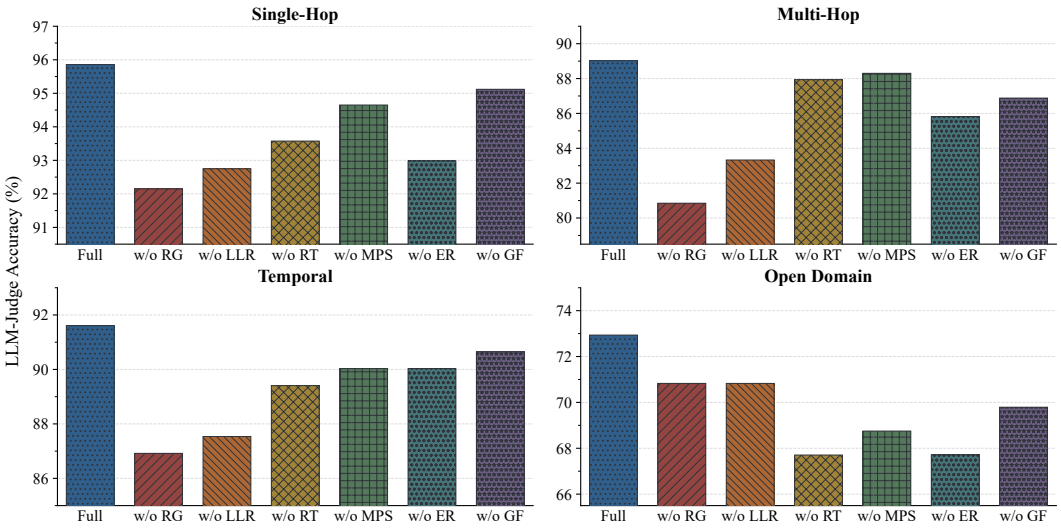


Figure 5: Ablation study on LoCoMo. RG: Relational Graph; LLR: Long-Level Relations; RT: Relation Types; MPS: Multi-Path Selection; ER: Evidence Recovery; GF: Guided Fallback.

ther variation produces no consistent degradation. These results support the default configuration $(H, B, K_i, \tau_R) = (3, 10, 5, 0.95)$ and demonstrate stable performance over broad parameter ranges. Sensitivity analyses for the remaining six hyperparameters are reported in Appendix C.3.

Ablation study. We ablate six components of `WeaveMem` on LoCoMo under controlled evaluation settings: Relational Graph (RG), Long-Level Relations (LLR), Relation Types (RT), Multi-Path Selection (MPS), Evidence Recovery (ER), and Guided Fallback (GF). Table 4 reports overall benchmark-specific score, and Figure 5 shows the results by question category. Removing any component reduces overall accuracy by 1.19–4.63 percentage points, with the largest drops caused by removing RG, LLR, and ER. At the category level, RG and LLR are most critical for multi-hop and temporal reasoning, whereas RT, MPS, ER, and GF contribute more strongly to open-domain questions. These results demonstrate that relational structure, applicability tracing, and evidence recovery play distinct and complementary roles in `WeaveMem` across varied question types and settings.

Configuration	Overall	Δ
`WeaveMem`	**92.29**	–
w/o RG	87.66	4.63 ↓
w/o LLR	88.57	3.72 ↓
w/o RT	90.06	2.23 ↓
w/o MPS	90.91	1.38 ↓
w/o ER	89.48	2.81 ↓
w/o GF	91.10	1.19 ↓

Table 4: Overall LoCoMo ablation study.

5 Conclusion

We introduced *applicability reasoning* as a distinct objective for long-term agent memory in complex multi-step interactive decision-making, requiring a system to determine not only which memories are connected, but how their relations jointly support the current query under varied task and interaction conditions. To address this objective, we proposed `WeaveMem`, which grounds typed and directional relations in atomic memories, projects them onto a compact experience graph, and constructs query-specific evidence chains through applicability tracing. Across four long-term memory benchmarks, `WeaveMem` consistently achieves the highest overall LLM-judge accuracy while maintaining a moderate answer-context size. Controlled evaluation further shows that its gains arise from constructing more complete and logically valid evidence chains rather than merely retrieving additional content. Together with the component ablations, these results demonstrate the value of making relational applicability explicit in long-term memory access.

AI USE STATEMENT

In this work, we used generative AI tools for none of the tasks with required disclosure. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, or interpret results. The remaining tasks with required disclosure to generate synthetic data sets, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, assist with translation, clean and reformat dataset, and support qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to edit a research paper to improve readability. All AI-suggested edits were reviewed by the authors to ensure that they preserved the intended meaning and did not alter any technical claims. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The methodology and algorithmic components of `WeaveMem` are described in Section 3. The datasets, baselines, evaluation protocol, model configurations, and implementation settings are documented in Section 4 and the appendix. The appendix additionally provides the prompts, complete hyperparameter settings, controlled applicability-evaluation protocol, and extended experimental results.

To facilitate reproducibility, the source code and configuration files are available in an anonymous repository at `https://github.com/WeaveMemory/WeaveMem`.

REFERENCES

Alibaba Cloud. Text embedding models. `https://www.alibabacloud.com/help/en/model-studio/embedding`, 2026. Accessed: 2026-09-13.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In Inês Lynce, Nello Murano, Mauro Vallati, Serena Villata, Federico Chesani, Michela Milano, Andrea Omicini, and Mehdi Dastani (eds.), *ECAI 2025 - 28th European Conference on Artificial Intelligence, 25-30 October 2025, Bologna, Italy - Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025)*, 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`.

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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL `https://proceedings.mlr.press/v267/gutierrez25a.html`.

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 Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 45836–45853. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.2125. URL `https://doi.org/10.18653/v1/2026.acl-long.2125`.

Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J. Taylor, and Dan Roth. Know me, respond to me: Benchmarking LLMs for dynamic user profiling and personalized responses at scale. In *Conference on Language Modeling*, 2025. URL `https://arxiv.org/abs/2504.14225`.

Dongming Jiang, Yi Li, Guanpeng Li, and Bingzhe Li. MAGMA: A multi-graph based agentic memory architecture for AI agents. *arXiv preprint arXiv:2601.03236*, 2026. doi: 10.48550/arXiv.2601.03236. URL https://arxiv.org/abs/2601.03236.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 25961–25970. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.1318. URL https://doi.org/10.18653/v1/2025.emnlp-main.1318.

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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, volume 33, pp. 9459–9474, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 13851–13870. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.747. URL https://doi.org/10.18653/v1/2024.acl-long.747.

OpenAI. GPT-4o mini: Advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/, 2024. Accessed: 2026-09-13.

OpenAI. Introducing GPT-4.1 in the API. https://openai.com/index/gpt-4-1/, 2025. Accessed: 2026-09-13.

Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *CoRR*, abs/2310.08560, 2023. doi: 10.48550/ARXIV.2310.08560. URL https://doi.org/10.48550/arXiv.2310.08560.

Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In Sean Follmer, Jeff Han, Jürgen Steimle, and Nathalie Henry Riche (eds.), *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST 2023, San Francisco, CA, USA, 29 October 2023- 1 November 2023*, pp. 2:1–2:22. ACM, 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. *CoRR*, abs/2501.13956, 2025. doi: 10.48550/ARXIV.2501.13956. URL https://doi.org/10.48550/arXiv.2501.13956.

Yiheng Shu, Saisri Padmaja Jonnalagedda, Xiang Gao, Bernal Jimenez Gutierrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. REMem: Reasoning with episodic memory in language agent. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=gmyq1saAh7.

Zihao Tang, Xin Yu, Ziyu Xiao, Zengxuan Wen, Zelin Li, Jiaxi Zhou, Hualei Wang, Haohua Wang, Haizhen Huang, Weiwei Deng, Feng Sun, and Qi Zhang. Mnemis: Dual-route retrieval on hierarchical graphs for long-term LLM memory. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 23914–23928. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.1096. URL https://doi.org/10.18653/v1/2026.acl-long.1096.

Mohammad Tavakoli, Alireza Salemi, Carrie Ye, Mohamed Abdalla, Hamed Zamani, and J. Ross Mitchell. Beyond a million tokens: Benchmarking and enhancing long-term memory in LLMs. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=y59hf5lrMn.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

Menglin Xia, Xuchao Zhang, Shantanu Dixit, Paramaguru Harimurugan, Rujia Wang, Victor Rühle, Robert Sim, Chetan Bansal, and Saravan Rajmohan. Memora: A harmonic memory representation balancing abstraction and specificity. *CoRR*, abs/2602.03315, 2026. doi: 10.48550/ARXIV.2602.03315. URL https://doi.org/10.48550/arXiv.2602.03315.

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. *arXiv preprint arXiv:2505.19549*, 2025a. doi: 10.48550/arXiv.2505.19549. URL https://arxiv.org/abs/2505.19549.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic memory for LLM agents. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025b. URL http://papers.nips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html.

Juwei Yue, Chuanrui Hu, Jiawei Sheng, Zuyi Zhou, Wenyuan Zhang, Tingwen Liu, Li Guo, and Yafeng Deng. HyperMem: Hypergraph memory for long-term conversations. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 35237–35254. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.1627. URL https://doi.org/10.18653/v1/2026.acl-long.1627.

Z.ai. GLM-5.2: Model overview. https://docs.z.ai/guides/llm/glm-5.2, 2026. Accessed: 2026-09-13.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 19724–19731. AAAI Press, 2024. doi: 10.1609/AAAI.V38I17.29946. URL https://doi.org/10.1609/aaai.v38i17.29946.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 15686–15702. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.971. URL https://doi.org/10.18653/v1/2023.emnlp-main.971.

APPENDIX CONTENTS

A Additional Method Details **14**

A.1 Notation and memory schemas 14

A.2 Relation semantics and grounding 15

A.3 Applicability tracing algorithm 16

B Experimental Setup and Reproducibility **17**

B.1 Datasets and category definitions 17

B.2 Evaluation metrics and repeated runs 18

B.3 Baseline implementation 19

B.4 Model and implementation configurations 22

B.5 Applicability evaluation protocol 22

B.6 Complete hyperparameter configuration 25

B.7 Component Ablation Protocol 25

C Additional Results and Analysis **27**

C.1 Complementary benchmark results 27

C.2 Additional efficiency results 30

C.3 Additional hyperparameter sensitivity 32

C.4 Complementary evaluation results 32

C.5 Qualitative evidence traces 33

C.6 Failure analysis 33

D Prompts **34**

D.1 Query analysis and relation prior 35

D.2 Trace expansion, pruning, and sufficiency evaluation 36

D.3 Answer generation and LLM-judge evaluation 40

A Additional Method Details

A.1 Notation and memory schemas

Notation and schemas. Let $\mathcal{M}$ denote the set of experiences. Each $m_i \in \mathcal{M}$ groups source turns from one session and is associated with atomic memories $\mathcal{L}(m_i)$, where $\mathcal{L} = \bigcup_{m_i \in \mathcal{M}} \mathcal{L}(m_i)$. For $\ell \in \mathcal{L}$, $\rho(\ell)$ is its parent experience, $z(\ell) \in \{\text{CORE}, \text{EPISODIC}, \text{KNOWLEDGE}\}$ is its type, and $\sigma(\ell)$ is its source-evidence list. Core memories describe stable attributes or traits; Episodic memories describe events; Knowledge memories record context-specific facts, including concrete information supplied by the assistant. Table A.1 summarizes their stored attributes. Temporal and entity metadata are stored explicitly where supported by the type; they need not be present or resolved in every record.

Unit	Stored attributes
Experience	`id`, `user_id`, `conversation_id`, `session_id`, `session_date`, `chat_ids`, `topic_subject`, `summary`, `tags`, `embedding`, `confidence`, `created_at`
CORE	`topic`, `subTopic`, `content`, `entityType`, `entityName`, `roleSpec`, `tags`
EPISODIC	`content`, `originalTimeExpression`, `mentionTime`, `eventTime`, `timePrecision`, `location`, `participants`, `eventTypeL1`, `eventTypeL2`, `roleSpec`, `context`, `tags`
KNOWLEDGE	`name`, `content`, `knowledgeType`, `context`, `importance`, `confidenceScore`, `status`, `statedBy`, `confirmationStatus`, `sourceChatIds`, `verbatimSpan`, `tags`

Table 5: Memory schemas.

Experience construction. A complete session, with turns in chronological order, is supplied to GLM-5.2 with participant names and a reference time. A single prompt first groups turns by topic consistency, contextual dependency, and intent progression, and then produces a topic, narrative summary, and tags for each group. The prompt requests that each turn belong to exactly one group and that each experience concern one participant. Participant membership is checked, but exhaustive coverage, disjointness, and contiguity of the groups are not independently enforced. The input unit is therefore a session, whereas an experience is a model-defined group within that session, not necessarily a contiguous interval. Each experience receives a new identifier and confidence 1.0; its base embedding is computed from its topic and summary using `text-embedding-v4`. Reprocessing a session rebuilds its memories rather than revising existing experience content in place.

Atomic extraction and evidence. Each experience is processed by three independent, type-specific extraction prompts. The evidence input is its original source turns. Each prompt returns a type-specific JSON array. The extractor attaches provenance metadata and normalizes Core tags; it does not enforce cross-type exclusivity or perform cross-type deduplication. Thus every record has one type, but different types may capture overlapping source information. The requested evidence list is $\sigma(\ell) = \{(chatId, quote)\}$, with each quote a contiguous verbatim fragment of a source utterance and a turn identifier when available. It refers to source spans, not entire sessions or internal-only evidence IDs. Knowledge records additionally include `sourceChatIds` and `verbatimSpan`. Experience and atomic extraction use GLM-5.2 with a single user message, no separate system message, non-streaming responses, a maximum of 20,000 output tokens, and temperature left at the endpoint default. JSON is parsed after generation rather than enforced through constrained decoding.

Relation representation. An atomic relation is $e^{\text{atom}} = (\ell_s, r, \ell_t, d_e, c_e)$, where r is its type, d_e is a model-generated natural-language description, and $c_e \in [0, 1]$ is its confidence. Direction is specified by the ordered endpoints, not by d_e. An experience-level edge is $e = (m_s, r, m_t, \mathcal{P}_e)$, with $m_s = \rho(\ell_s)$ and $m_t = \rho(\ell_t)$. Here $\mathcal{P}_e$ contains the retained supporting atomic relations: their relation identifiers, endpoint identifiers, descriptions, endpoint evidence phrases, and confidences. These identifiers permit tracing back to $\sigma(\ell_s)$ and $\sigma(\ell_t)$; the evidence phrases generated during relation induction are not themselves copies of the complete source-span lists.

Table 6: Relation semantics, read from the stored source to the stored target. Only CONTRADICTS is symmetric.

Relation	Source → target interpretation
SUPPORTS	Concrete evidence supports a claim.
CONTRADICTS	Two claims about the same subject are incompatible under comparable conditions.
EXPLAINS	A reason or necessary background explains a fact, event, or state.
CAUSES	An explicit direct cause or action produces an outcome.
PRECEDES	An earlier stage precedes a dependent later stage of the same bounded process; mere chronological order is insufficient.
ENABLES	A prerequisite, resource, capability, or opportunity makes an action or outcome possible.
UPDATES	An earlier state, plan, belief, or value is followed by its later replacement, correction, realization, or progress.

A.2 RELATION SEMANTICS AND GROUNDING

Relation inventory. The relation set $\mathcal{R}$ contains the seven types listed in Table 6. Only CONTRADICTS is symmetric; its endpoints are canonically ordered instead of storing both orientations. SIMILAR is not a stored relation: similarity is used for candidate retrieval, not as evidence of a typed dependency.

Candidate generation. Candidate atomic pairs are drawn from different experiences in the same conversation. They may cross sessions and participants; there is no fixed candidate time window. For each anchor, candidates are ranked by IDF-weighted tag overlap, positive-score BM25, and dense cosine similarity. Each channel is ranked without a preliminary top-k cutoff. If $\mathcal{H}_{ij}$ is the set of channels in which ℓ_j appears for anchor ℓ_i, fusion uses

$$s_{\text{pair}}(\ell_i, \ell_j) = \sum_{c \in \mathcal{H}_{ij}} \frac{1}{60 + \text{rank}_c(\ell_j \mid \ell_i)},$$

with ranks starting at zero. Only the top 10 fused candidates per anchor are retained. Duplicate discoveries of an unordered pair are merged, self-pairs and same-experience pairs are excluded, and the relation model determines the semantic orientation rather than inheriting the retrieval order.

Relation induction. Each uncached candidate pair receives GLM-5.2 judgment, using a JSON-only system instruction and a pair-specific user prompt, with a 700-token output cap, non-streaming responses, thinking disabled. The prompt includes the conversation and participant context, confidence threshold, and both endpoints' identifiers, types, contents, tags, and session metadata. Type-specific fields add Core entity and topic information, Episodic context, event and mention times and participants, or Knowledge name and context.

Grounding. For CAUSES, PRECEDES, ENABLES, and UPDATES, a relation is also rejected if both event-time starts are parseable and the source is later than the target. Equal times are permitted; missing or unparseable times do not independently reject an edge. Tag-based guards reject CONTRADICTS proposals with disjoint capitalized-tag entity sets but a shared category tag. For CAUSES, CONTRADICTS, PRECEDES, and UPDATES, proposals whose endpoints both mention a project but lack a shared non-generic identifying tag are also rejected. The acceptance rule is

$$\text{Accept}(e) = \mathbf{1}[c_e \geq \tau_R]\, g_{\text{model}}(e)\, g_{\text{rule}}(e).$$

These checks combine model judgments with limited deterministic guards. The default is $\tau_R = 0.95$. The threshold sensitivity study considers $\{0.60, 0.70, 0.80, 0.90, 0.95, 0.98, 0.99\}$.

Projection. Accepted atomic relations are lifted through ρ and merged by parent endpoints and relation type. Intra-experience edges and self-loops are omitted, and cross-conversation projection is disallowed. Different relation types between the same experiences remain separate, even though each atomic pair receives at most one type. Supporting records in $\mathcal{P}_e$ are deduplicated by atomic

Table 7: Default online tracing parameters. C_i, C_f are post-fusion candidate counts; K_i, K_f are seed counts after reranking.

Parameter	Value	Meaning
(C_i, K_i)	$(20, 5)$	Initial retrieval
(C_f, K_f)	$(50, 10)$	Fallback retrieval
H	3	Maximum trace length in hops
P	5	Maximum pruned traces per seed
B	10	Unique-experience budget
τ_{fact}	0.5	Atomic-evidence score threshold
K_{fact}	10	Global atomic-evidence cap
κ	60	RRF constant; ranks start at zero
(k_1, b)	$(1.5, 0.75)$	BM25 parameters; no lemmatization

endpoints and case-normalized description and evidence text; the first occurrence is retained in construction order, without a fixed provenance cap. Distinct descriptions are concatenated, and the projected confidence is the maximum over retained supporting records. The graph is constructed offline and held fixed during query-time tracing.

A.3 Applicability tracing algorithm

Query analysis and seed retrieval. A rule-based classifier returns a question type t_q, a preferred relation set $\mathcal{R}_q$, and direction guidance g_q. Its categories are *when*, *how*, *why*, *where*, *who*, *which*, *what*, *yes/no*, and *other*; it does not use an LLM or separately extract entities, temporal constraints, or subquestions. The relation prior informs controller decisions and trace scoring but is not a hard edge filter. Within the conversation, experience candidates are retrieved by dense cosine similarity using `text-embedding-v4` and by BM25, fused by RRF, and reranked with `qwen3-rerank`. Dense and reranking texts contain the experience topic, summary, and type-specific projections of its atomic memories; the BM25 text also includes experience tags. Candidate experiences are unique by identifier, with no same-session quota. The initial pass retains $C_i = 20$ fused candidates and $K_i = 5$ reranked seeds. Table 7 collects the default tracing parameters.

Trace expansion and pruning. Expansion is controller-directed rather than a fixed breadth-first or depth-first search. For each seed, the state maintains collected experiences, visited nodes, frontier edges, and node-simple traces. The controller sees the query, question type and relation guidance, collected experience text, and eligible frontier edge cards containing relation type, description, traversal direction, and whether the destination is one of the retrieved seeds. The destination node's own text is hidden until the edge is selected. All eligible frontier edges are exposed, rather than a fixed top-k subset. One call may select multiple edges, report missing evidence, or declare the collected evidence sufficient. Selected destinations are added to the state, visited nodes and the frontier are updated, and cycles within a trace are excluded. Reverse traversal is allowed without reversing the stored relation's meaning. Expansion stops on a sufficiency decision, no selected edge, no newly added node, or the hop limit H. The pruning controller then selects at most P traces per seed, favoring query-relevant, complementary evidence and shorter traces when coverage is equivalent.

Sufficiency and fallback. After per-seed pruning, a sufficiency call examines the deduplicated experience text represented by the selected traces and seeds. If insufficient, retrieval is repeated with $(C_f, K_f) = (50, 10)$ and tracing is rerun. This replaces the initial seeds and traces rather than appending to them; the new seed set may still overlap the old one. Fallback occurs at most once, without a second sufficiency call.

Trace scoring and admission. Let $R(q, x)$ denote the `qwen3-rerank` relevance score for text x. For a trace $\pi = (m_0, e_1, m_1, \ldots, e_h, m_h)$, let $x(m)$ be its rendered experience text: topic and summary, together with any available intent annotations. With $\|$ denoting concatenation in traversal

order, the node, description, and type components for $h > 0$ are

$$S_{\text{node}}(\pi) = R\big(q, x(m_0) \| \cdots \| x(m_h)\big),$$
$$S_{\text{desc}}(\pi) = R\big(q, d_{e_1} \| \cdots \| d_{e_h}\big),$$
$$S_{\text{type}}(\pi) = \frac{1}{h} \sum_{j=1}^{h} \mathbf{1}[r(e_j) \in \mathcal{R}_q].$$

Here d_{e_j} is the stored description of the traversed experience edge, aggregated from its supporting atomic relations. The fourth component, $S_{\text{ctrl}}(\pi)$, is the pruning controller's path-value score. For zero-hop candidates, $S_{\text{type}} = S_{\text{desc}} = 0.5$ and the seed reranking score supplies S_{ctrl}. Components are clipped to $[0, 1]$, with missing controller values set to 0.5. The combined score is

$$S(\pi) = w_n S_{\text{node}}(\pi) + w_t S_{\text{type}}(\pi) + w_d S_{\text{desc}}(\pi) + w_c S_{\text{ctrl}}(\pi).$$

Candidates are deduplicated by traversal sequence and sorted by decreasing $S(\pi)$, breaking ties by fewer nodes and then candidate identifier. Let U be the already admitted experience set and $V(\pi)$ the experiences in a candidate trace. A trace is admitted exactly when

$$V(\pi) \setminus U \neq \varnothing \quad \text{and} \quad |U \cup V(\pi)| \leq B.$$

Shared experiences count once, zero-gain traces are skipped, and traces are never truncated to fill the remaining budget. Zero-hop candidates compete under the same rule; not every seed is guaranteed to survive.

Atomic evidence and context assembly. For the admitted experiences $\mathcal{M}_q^\star$, all attached atomic memories with nonempty rendered content enter one global reranking pool. If $y(\ell)$ is the type-specific text projection of a memory, its score is $S_{\text{fact}}(\ell) = R(q, y(\ell))$. Core projections contain participant, entity, topic, subtopic, and content; Episodic projections contain participant, content, context, and temporal fields; Knowledge projections contain name and content. Evidence recovery keeps the highest-scoring $K_{\text{fact}} = 10$ memories among those with $S_{\text{fact}} \geq \tau_{\text{fact}} = 0.5$; there is no generative fact-selection controller. Each admitted experience appears once, in first-admission order, with its retained facts grouped beneath it in reranking order. Stored source-evidence spans are appended to the selected atomic text when available. The resulting $\mathcal{C}_q$ supplies experience text and atomic evidence to the answerer together with the query. Algorithm 1 summarizes the execution order.

B Experimental Setup and Reproducibility

This section specifies the benchmark coverage, evaluation protocol, baseline conditions, model assignments, and configuration required to reproduce the reported experiments.

B.1 Datasets and category definitions

We evaluate four long-term memory benchmarks using their official evaluation sets. LoCoMo (Maharana et al., 2024) evaluates reasoning over multi-session conversations through single-hop, multi-hop, temporal, and open-domain questions. LongMemEval-S (Wu et al., 2025) covers knowledge updates, cross-session reasoning, temporal reasoning, and session-level information extraction. PersonaMem-32K (Jiang et al., 2025) evaluates personalized responses grounded in evolving user information. BEAM-1M (Tavakoli et al., 2026) evaluates diverse memory capabilities over histories approaching one million tokens. Table 8 records the exact evaluated inputs, counts, category tallies, and length statistics.

Preprocessing and truncation. For LoCoMo, `session_N` keys are numerically ordered and mapped to `D1`, `D2`, and so forth; natural-language session timestamps are normalized to `YYYY-MM-DD HH:mm:ss`, with unparseable values retained, and each turn is rendered as `[dia_id] speaker: text`. The evaluated file contains no empty-text turns, and the loader performs neither turn deduplication nor system-message insertion. For LongMemEval, every question is loaded into an isolated store, released session order and duplicate filler sessions are retained, dates are normalized from `YYYY/MM/DD (Day) HH:MM`, and positional identifiers `D1`, `D1:1`,

Algorithm 1 Applicability tracing

Require: Query q, graph $\mathcal{G} = (\mathcal{M}, \mathcal{E})$, atomic memories $\mathcal{L}$
Require: $C_i, K_i, C_f, K_f, H, P, B, K_{\text{fact}}, \tau_{\text{fact}}$ and score weights
Ensure: Evidence context $\mathcal{C}_q$
$(t_q, \mathcal{R}_q, g_q) \leftarrow \textsc{AnalyzeQuery}(q)$
$\mathcal{S}_q \leftarrow \textsc{RetrieveSeeds}(q, \mathcal{M}, C_i, K_i)$
$\Pi_q \leftarrow \textsc{TraceAndPrune}(q, \mathcal{S}_q, \mathcal{G}, t_q, \mathcal{R}_q, g_q, H, P)$
$\Pi_q \leftarrow \Pi_q \cup \{(m) : m \in \mathcal{S}_q\}$
if $\neg\textsc{Sufficient}(q, \textsc{ExperienceText}(\Pi_q))$ **then**
    $\mathcal{S}_q \leftarrow \textsc{RetrieveSeeds}(q, \mathcal{M}, C_f, K_f)$
    $\Pi_q \leftarrow \textsc{TraceAndPrune}(q, \mathcal{S}_q, \mathcal{G}, t_q, \mathcal{R}_q, g_q, H, P)$
    $\Pi_q \leftarrow \Pi_q \cup \{(m) : m \in \mathcal{S}_q\}$
end if
Deduplicate candidates by traversal sequence and compute $S(\pi)$
Sort by decreasing score, then fewer nodes, then candidate identifier
$U \leftarrow \varnothing$; $O \leftarrow [\,]$; $\Pi_q^\star \leftarrow \varnothing$
for all π in the sorted candidate list **do**
    if $V(\pi) \setminus U \neq \varnothing$ **and** $|U \cup V(\pi)| \leq B$ **then**
        $\Pi_q^\star \leftarrow \Pi_q^\star \cup \{\pi\}$
        Append nodes of π not in U to O in traversal order
        $U \leftarrow U \cup V(\pi)$
    end if
end for
$\mathcal{M}_q^\star \leftarrow U$
$\mathcal{F}_q \leftarrow \textsc{RecoverEvidence}(q, \mathcal{M}_q^\star, \mathcal{L}, K_{\text{fact}}, \tau_{\text{fact}})$
$\mathcal{C}_q \leftarrow \textsc{AssembleContext}(O, \mathcal{F}_q)$
return $\mathcal{C}_q$

and so forth replace source identifiers that could reveal answer or abstention labels. Roles must be exactly `user` or `assistant`. Thirteen histories each contain one repeated source-session identifier, and none is removed. For PersonaMem, whitespace is collapsed, identifiers and parent/type fields are normalized, answer options are removed from retrieval-only queries where required, and system messages delimit sessions; the initial persona system message is retained once, repeated persona copies are excluded, `User:`/`Assistant:` prefixes are removed before speaker mapping, timestamps are set to null, and no remaining turns are deduplicated or filtered. Its cutoff is exclusive, so each query sees only messages preceding its recorded endpoint; this is benchmark-time filtering rather than context-overflow truncation. The BEAM parquet contains ten nested sessions per history, 37,315 user and 37,315 assistant turns, and no system, empty, or exactly duplicated turns. Raw turn identifiers restart across sessions in four histories, so provenance is scoped as `D<session>:<source-id>`; all 32,668 stored Mid records match the normalized `Month-DD-YYYY` to midnight-`YYYY-MM-DD HH:mm:ss` session dates, although 12 generated Mid provenance strings are malformed. No benchmark path applies token-based clipping after rendering the answer prompt: raw-history cutoffs and retrieval budgets select evidence but do not silently shorten a completed prompt; the completed prompt is sent unchanged, and an over-limit provider error follows the normal retry/failure path rather than producing a truncated evaluation example. BEAM additionally rejects empty responses and responses whose `finish_reason` is not `stop`.

B.2 Evaluation metrics and repeated runs

Run protocol. The primary quality metric is LLM-judge answer accuracy: each judged answer contributes one correctness indicator, and the reported accuracy is the mean of these indicators as a percentage. Category and overall scores use the same aggregation. Each benchmark is evaluated over five full runs and each run redoes extraction, graph construction, and retrieval. No random seeds are used anywhere; the only seed-like setting is `PYTHONHASHSEED=0` in the LoCoMo run environment, which fixes hash iteration order and does not affect sampling. Failed API calls are retried at the transport layer (timeout 400 s, up to 5 attempts; reranker 3 attempts); an answer that is

Table 8: Exact dataset records used in the reported evaluations.

Dataset	Release/Split	#Conversations	#Sessions	#Queries	Category Counts	History Length Statistics
LoCoMo	GitHub `snap-research/locomo`.	10	272	1,986 raw; 1,540 evaluated	Evaluated: MH (multi-hop) 282; Temp (temporal) 321; OD (open-domain) 96; SH (single-hop) 841. The 446 adversarial examples are excluded.	Raw per history: 369–689 turns (mean 588.2, median 646) and 43,587–89,736 characters (mean 72,675.6). Retrieved answer prompts: 2,242–6,667 API tokens (mean 4,478.71, median 4,439, P95 5,840.4).
LongMemEval	HuggingFace `xiaowu0162/longmemeval-cleaned`.	500	23,867	500 / 500	SS-U (single-session-user) 70; SS-A (single-session-assistant) 56; SS-P (single-session-preference) 30; multi-session 133; temporal-reasoning 133; knowledge-update 78.	Raw per history: 396–616 turns (mean 493.5, median 491, P95 545.1) and 454,980–513,953 characters (mean 489,297.71, median 489,793.5, P95 503,398.15).
PersonaMem	HuggingFace `bowen-upenn/PersonaMem-v1`;	37	195	589 / 589 (222 distinct cutoff prefixes)	GNS (generalizing to new scenarios) 57; PAR (provide preference-aligned recommendations) 55; RUSF (recall user-shared facts) 129; RFMU (recalling facts mentioned by the user) 17; RPU (recalling the reasons behind previous updates) 99; SNI (suggest new ideas) 93; TFPE (track full preference evolution) 139.	Raw per history: 116–238 messages (mean 174.65). Query-visible exclusive-cutoff prefixes: 96–237 messages per query (mean 167.12).
BEAM	GitHub `mohammadtavakoli78/BEAM`, `chats/1M/1--35`.	35	350	700 / 700 per run;	Seventy each.	Raw per history: 1,556–3,008 turns (mean 2,132.29, median 2,088, P95 2,908.2) and 2,842,702–7,376,557 characters (mean 4,573,524.43, median 4,348,844, P95 6,902,900.7).

still empty after retries aborts the run, which resumes from checkpoints—no question is silently discarded. Judge failures follow the same halt-and-resume policy on LoCoMo and LongMemEval; on BEAM a malformed judge response scores the affected rubric item 0.0 and the question is retained; on PersonaMem an unparseable response is scored incorrect and remains in the denominator. For run r with benchmark score a_r, reported multi-run summaries are the unweighted mean $\bar{a} = \frac{1}{5}\sum_{r=1}^{5} a_r$ and the sample standard deviation $s_a = \sqrt{\frac{1}{4}\sum_{r=1}^{5}(a_r - \bar{a})^2}$ (ddof $= 1$) computed across the five run-level scores, never across individual question outcomes.

B.3 Baseline implementation

We compare WeaveMem with eight representative long-term memory systems. Mem0 (Chhikara et al., 2025) stores fact-based memories. A-Mem (Xu et al., 2025b) dynamically organizes memories through evolving associations. Graphiti (Rasmussen et al., 2025) constructs a temporal knowledge graph. REMem (Shu et al., 2026) performs episodic retrieval with reasoning. MemGAS (Xu et al., 2025a), Memora (Xia et al., 2026), HyperMem (Yue et al., 2026), and EverMemOS (Hu et al., 2026) represent multi-granularity, hierarchical, hypergraph-based, and self-organizing memory designs, respectively. A local Full Context control directly supplies the available history to the answer model subject to its context limit. Table 10 lists each baseline's implementation source, recorded version identifier, core retrieval configuration, and context budget; supporting parameters and benchmark-specific differences are detailed below. The settings are drawn from the reproduced code, configuration files, and frozen experiment artifacts, with frozen run records taking precedence over current defaults. For cross-system comparability, the common answering setup uses the shared answerer `gpt-4o-mini` at temperature 0 rather than each baseline's native answerer. Judging uses `gpt-4.1-mini` for LoCoMo , LongMemEval-S and BEAM-1M, and deterministic answer parsing for PersonaMem-32K.

Metric	Unit	Aggregation
LoCoMo LLM-judge accuracy	fraction (%)	Flat mean over all 1,540 scored questions; per-category means over 282 (multi-hop), 321 (temporal), 96 (open-domain), 841 (single-hop); category-5 adversarial questions are excluded by the scorer
LongMemEval-S LLM-judge accuracy	fraction (%)	Overall over 500 questions (470 answerable + 30 abstention); abstention reported separately
PersonaMem-32k MCQ accuracy	fraction (%)	Correct / completed over 589 questions from 37 shared contexts (per type: 57, 55, 129, 17, 99, 93, 139)
BEAM nugget score	0–1 ($\times$100)	Per-run mean over 700 questions (10 categories $\times$ 70); per-category means over 70 each
Answer-overlap F1 / ROUGE-1/2/L / BLEU-1–4	ratio	Macro mean over questions (LoCoMo N=1,540; LongMemEval N=500; BEAM N=700)
Native-unit exact-turn Recall-all@K	fraction (%)	Fraction of questions whose top-K native retrieval units jointly cover all gold evidence turns; denominators 1,536 (LoCoMo), 470 (LongMemEval answerable), 625 (BEAM)
Answer prompt tokens	tokens / question	Mean over the benchmark's questions

Table 9: Evaluation metrics, units, and aggregation protocols.

Reimplementation and deviation. For EVEROS, HYPERMEM, and MEMGAS we retain the retrieval, indexing, rerank, and context-assembly pipelines of their repository implementations. The other memory systems include local retry and model/API compatibility adaptations intended to preserve the main algorithmic flow, together with the benchmark-specific assembly settings recorded here; we disclose their effects on prompts, answer parsing, and token accounting rather than treating every run as an unmodified upstream execution. No additional reimplementation of an unavailable component was established in the audit, and Full Context is the local control implementation. The shared answerer receives only the memories returned and assembled for the query by each baseline; Full Context instead supplies conversation history under the policy below. In the additional-baseline artifacts the answer-model identifier is generally `gpt-4o-mini`, while some manifests use `gpt-4o-mini`; the common answerer temperature does not imply that baseline-internal extraction models all ran at temperature 0.

EVEROS exposes two answer-model configuration layers (`gpt-4.1-mini` as the framework default and `gpt-4o-mini` as the benchmark answerer); the benchmark harness reads the latter, so the answerer is `gpt-4o-mini`. Agentic retrieval uses 50 dense candidates, 50 sparse candidates, RRF $k = 40$, cluster top-$k = 10$, a round-1 top-50 reduced to top-10 after reranking, a round-2 cap of 40, and 3 multi-queries, with `Qwen/Qwen3-Embedding-4B` embeddings and the `Qwen/Qwen3-Reranker-4B` reranker; it returns 10 episodes without token-length truncation.

HYPERMEM uses a three-level hypergraph (topics $\rightarrow$ episodes $\rightarrow$ facts), RRF constant $K = 60$, and `text-embedding-v4` embeddings. The `qwen3-rerank` cross-encoder is available, which we retain for cross-system uniformity. Memory extraction uses `glm-5.2` at temperature 0, output limit 16,384, and thinking disabled. Per-stage defaults are topic top-10, episode top-20, and fact top-30; the frozen BEAM-1M run overrides episode top-k to 25. No token truncation is applied before the answerer.

MEMGAS constructs a four-granularity graph over sessions, turns, summaries, and keywords. Extraction uses `glm-5.2` at temperature 0, `max_tokens` $= 500$, and thinking disabled; embeddings use `text-embedding-v4`. Retrieval combines an entropy-weighted soft router over the four granularities with personalised PageRank on an igraph graph. The exercised defaults are `num_seednodes` $= 15$, `mem_threshold` $= 30$, `n_components` $= 2$, `anchor_sessions` $= 5$, `damping` $= 0.1$, and `temp` $= 0.1$. It returns top-3 snippets, each combining a session block,

System	Source	Version	Config.
EVEROS	github.com/EverMind-AI/EverOS.git	d07cddc4; package 1.2.3.	Agentic hybrid retrieval; dense/sparse top-50; reciprocal-rank fusion (RRF) $k = 40$; Qwen3 embedding and reranking.
HYPERMEM	github.com/EverMind-AI/HyperMem.git	15c70090.	Topic/episode/fact hypergraph; RRF $K = 60$; glm-5.2 extraction.
MEMGAS	github.com/Applied-Machine-Learning-Lab/ICLR2026_MemGAS.git	c2d4e9fd.	Four-granularity graph; entropy-weighted routing and personalised PageRank; no reranker.
MEM0	github.com/mem0ai/mem0	760dca6f.	Dense fact memory; LoCoMo threshold 0.1, reranking off, expired memories excluded.
A-MEM	github.com/agiresearch/A-mem	ceffb860.	GLM query rewriting; cosine top-20 seeds plus one-hop linked neighbors; deduplicate by memory ID.
GRAPHITI	github.com/getzep/graphiti	899cb40d.	Edge/node/community retrieval; cross-encoder and RRF routing vary by benchmark.
REMEM	github.com/intuit-ai-research/REMem	d87840be.	Session episodic gist; retrieval/linking/QA top-k: $200/5/10$; 3 agent steps; DSPy reranking.
MEMORA	github.com/microsoft/Memora	dec3f8f2.	PromptedPolicy; 5 effective steps; query/cue/hybrid top-k: $30/20/10$.
MAGMA	Upstream MAGMA with local benchmark adapters.	467cb70b; adapted.	Turn/session multi-graph; RRF $k = 60$; original-query top-15 plus decomposed subquery retrieval.
Full Context	Local evaluation harness.	—	Complete conversation history; no memory construction, embeddings, retrieval, or reranking.

Table 10: Reproduced baseline implementations and exercised settings.

its summary, and its keywords, and applies a $4{,}000$-token answer-output limit at temperature 0 that is not a memory-context truncation limit.

MEM0 uses dense fact retrieval with top-10 memories. Its LoCoMo run uses a retrieval threshold of 0.1, disables reranking, and excludes expired memories. The LoCoMo extractor is glm-5.2 with temperature 0.1, top-$p = 0.1$, a $2{,}000$-token output limit, and thinking disabled; text-embedding-v4 produces $1{,}024$-dimensional embeddings.

A-MEM rewrites queries with GLM, selects the cosine-similarity top-20 seed memories, appends all real one-hop linked neighbors, and deduplicates by memory ID without recursive expansion, so the seed limit does not impose a final memory-count cap.

GRAPHITI retrieves the top-10 edges, nodes, and communities separately, returning at most 30 items; communities participate in retrieval. LoCoMo and PersonaMem-32K use a cross-encoder for edges and RRF for nodes and communities, whereas BEAM-1M uses RRF on all three routes. LoCoMo's three top-10 lists are deterministic subsets of frozen per-route top-20 results. PersonaMem-32K isolates snapshots and BEAM-1M isolates conversations. The LongMemEval-S record specifies glm-5.2 with a $4{,}096$-token output limit and text-embedding-v4 with $1{,}024$ dimensions.

REMEM uses session episodic gists, retrieval/linking/QA top-k values of $200/5/10$, 3 agent steps, DSPy reranking, damping 0.5, and passage weight 0.05. Its graph is unidirectional and contains

facts and similar-passage nodes; duplicate node IDs retain their maximum score, and results are sorted by score. The synonymy threshold is 0.8 with top-k $= 2{,}047$. Context assembly returns at most 20 items for LoCoMo and 60 for LongMemEval-S, PersonaMem-32K, and BEAM-1M; empty recall remains empty. BEAM-1M ingestion packs at most $48{,}000$ characters without splitting a turn group, and its records specify a $65{,}536$-token extraction-output limit and $1{,}024$-dimensional embeddings.

MEMORA uses PromptedPolicy with 5 effective steps, query/cue/hybrid top-k values of $30/20/10$, query and BM25 thresholds of 0.4, an update threshold of 0.8, and cosine similarity. Factual, episodic, cue, and hybrid memories are enabled; query enhancement, LLM filtering, and relaxed frontier are disabled. The effective five-step limit differs from the four-step YAML value that was not passed through to the policy. Per-query top-k does not cap the final context because expansion and re-query can accumulate items. BEAM-1M uses RRF weights 2:1:1. The `use_segments_as_episodic` flag is true for LoCoMo and PersonaMem-32K (raw conversation segments as episodic values) and false for LongMemEval-S and BEAM-1M (generated episodic summaries with the original text retained separately); both modes retain episodic and factual memory generation, and the PersonaMem-32K adapter overrides this flag when merging its settings into the LongMemEval configuration.

For EVEROS, HYPERMEM, and MEMGAS, scoring follows the shared per-benchmark protocol: `gpt-4.1-mini` at temperature 0 for BEAM-1M, `gpt-4.1-mini` at temperature 0 for LoCoMo and LongMemEval-S, and the deterministic parenthesised-first set-equality parser with no judge for PersonaMem-32K.

On the Full Context baseline. LoCoMo preserves the complete conversation, including speaker, text, date, and image-caption fields. LongMemEval-S stably sorts sessions by date and includes the complete sessions. PersonaMem-32K uses messages before the exclusive cutoff, preserving original roles, content, and repeated system-persona messages. These three target datasets fit without truncation. BEAM-1M retains the earliest continuous prefix, ending only after a complete user–assistant exchange and reserving $4{,}096$ output tokens within the $128{,}000$-token window (FullContext-windowed). This input-window policy is distinct from whether generation actually reached an output-token limit, which was not verified here.

B.4 MODEL AND IMPLEMENTATION CONFIGURATIONS

Table 11 lists every model invoked to produce the reported results. Three public providers are involved: Zhipu GLM-5.2 (Z.ai, 2026) for memory construction and applicability tracing, Alibaba Cloud DashScope (`text-embedding-v4` (Alibaba Cloud, 2026)) for embedding and reranking, and OpenAI (`gpt-4o-mini` (OpenAI, 2024) answers, `gpt-4.1-mini` (OpenAI, 2025) judging) for the GPT-family answer and judge models. All chat calls use the OpenAI Chat Completions protocol.

B.5 APPLICABILITY EVALUATION PROTOCOL

Scope and shared inputs. We evaluate query-conditioned evidence selection using two fixed-graph diagnostics. The LoCoMo panel contains 252 official questions from ten conversations: 200 distributed-reference candidates, 50 single-memory reference-coverage controls, and two supplemental candidates. Each question uses the complete frozen graph for its conversation, the same two dense-ranked initial seeds for graph-based methods, and frozen query–memory semantic scores.All 252 questions underwent independent AI review and AI adjudication, followed by source-based human review of the complete set rather than only AI disagreements. Human reviewers inspected the questions, reference answers, original conversation turns, memory facts, citations, and proposed evidence labels without seeing method identities, arm assignments, or system outcomes. An AI-assisted label was retained only after human verification and was revised whenever it conflicted with the source evidence. The reported results therefore use human-reviewed final annotations rather than unverified AI outputs. We report AI–AI agreement only as a consistency diagnostic and do not present it as human agreement.

We construct a fixed-graph chain-recovery diagnostic from 400 MQuAKE-TZhong et al. (2023) cases: 303 two-fact, 95 three-fact, and 2 four-fact chains. It evaluates evidence selection within a

Component	Model	Provider / version	Function
Atomic memory extraction	`glm-5.2`	Zhipu GLM	Session-level atomic fact extraction
Relation induction (pair judge)	`glm-5.2`	Zhipu GLM	One JSON verdict per candidate pair
Graph navigation / sufficiency / path selection	`glm-5.2`	Zhipu GLM	Retrieval-time control calls
Embedding	`text-embedding-v4`	Alibaba Cloud DashScope	Dense vectors for memories and queries
Reranker	`qwen3-rerank`	Alibaba Cloud DashScope	Cross-encoder reorder of fused candidates
Answer, LoCoMo	`gpt-4o-mini`	OpenAI	QA answer from selected memories
Answer, LongMemEval-S	`gpt-4o-mini`	OpenAI	QA answer
Answer, PersonaMem-32K	`gpt-4o-mini`	OpenAI	MCQ answer
Answer, BEAM-1M	`gpt-4o-mini`	OpenAI	Nugget-style answer
Judge, LoCoMo	`gpt-4.1-mini`	OpenAI	CORRECT/WRONG
Judge, LongMemEval-S	`gpt-4.1-mini`	OpenAI	CORRECT/WRONG
Judge, BEAM-1M	`gpt-4.1-mini`	OpenAI	Rubric item and event-ordering grading
Judge, PersonaMem-32K	*none*	—	Deterministic parenthesised-first set-equality parser

Table 11: Component–model mapping for the reported results.

supplied graph and from a supplied starting node; it is not an end-to-end retrieval or seed-discovery benchmark. Each labeled source triple (s, p, o) becomes one fact node containing the text "$s\ p\ o$." Consecutive facts are connected according to the reference-chain order. The stored edge label is the native MQuAKE predicate of the corresponding source fact. It is not mapped to, and should not be interpreted as, one of WeaveMem's seven inter-memory relation types.

Distractors are sampled from the dataset-wide fact bank using shared entities or predicates. We exclude facts from the same case, exact duplicates of reference triples, and facts containing the reference answer or its aliases. Entity-sharing distractors may be connected to reference nodes according to their shared subject or object. No additional reference-to-reference edges or distractor-to-distractor edges are introduced.

The reference chain determines the reference nodes, consecutive edges, traversal order, and evaluation annotations. Its first node is always supplied as the initial seed. The answer and its aliases are used only to exclude answer-containing distractors. Thus, the diagnostic remains conditional on a reference-constructed graph and a supplied chain start.

The original graphs used systematic identifiers that distinguished reference and distractor records. We therefore repeated the complete evaluation after replacing every node, edge, and path identifier with an opaque random identifier, independently shuffling candidate presentation order, and removing synthetic date-based ordering. The questions, textual facts, graph topology, supplied seeds, semantic scores, models, prompts, and budgets were otherwise unchanged.

Across three independent permutations, Full WeaveMem obtained answer accuracies of 89.25%, 90.25%, and 88.75% (mean 89.42%), compared with 89.75% in the original evaluation. Mean complete-reference-trace recovery was 94.17%, and direction-aware reference-edge recall was 94.96%. The stability of these results indicates that systematic identifier and ordering patterns do not explain the reported absolute performance. This audit does not remove the supplied reference-chain seed.

All constructed graphs and their relation and direction annotations underwent source-based human review. Review packets contained the endpoint facts, source triples, stored relation, and traversal direction, but omitted method and arm names, system answers, scores, reference-versus-distractor

flags, and original structured identifiers. Records were presented under opaque aliases and in random order. Human reviewers assessed the source evidence rather than being asked merely to confirm an AI judgment. When an automatic annotation conflicted with the source-based human assessment, the human assessment determined the final annotation.

This review verifies the fidelity of the constructed graph and its reference annotations. It does not convert the graph-supplied predicate into a relation predicted by the controller, nor does it establish a mapping to WeaveMem's seven relation types.

The conversational panel compares four selection procedures; the MQuAKE panel compares Full WeaveMem and WeaveMem w/o Applicability.

Selection procedures. The experimental arms differ as follows.

- **Semantic Retrieval.** Memories are ranked by their frozen query–memory semantic scores, without graph traversal or relation information. The requested selection count equals the number of memories retained by Full WeaveMem for the same question, after which the semantic selection is independently packed under the context-token limit. The token limit reduces the semantic selection on the remaining two. An empty Full selection therefore also gives this count-matched baseline an empty selection. This arm makes no selector LLM calls.
- **Untyped Graph Traversal.** From each seed, the method takes at most three step to the neighbour with the highest frozen query–memory semantic score. Edge types, descriptions, and stored directions do not affect selection. A seed without a usable neighbour is skipped. The resulting routes are ranked by mean node semantic score and passed to the shared path-packing procedure.
- **WeaveMem w/o Applicability.** It retains the same LLM-based selection pipeline as Full WeaveMem, including the frozen memory graph, initial seeds, model configuration, and context-packing procedure. It removes explicit query-conditioned relation-applicability assessment while retaining content relevance, redundancy reduction, and answer-coverage assessment. Both variants use the same configured inference and answer-context budgets.
- **Full WeaveMem.** The adapter obtains a question-type strategy and jointly evaluates the seed memories for sufficiency. When the seeds are jointly sufficient and fit the budgets, they are retained as singleton paths. Otherwise, each seed initiates iterative navigation. At each step, the controller receives the collected memories and candidate-edge types, traversal directions, and descriptions. The topic and summary of an unvisited neighbour are not displayed before its edge is selected. Multiple edges may be selected in one navigation call. Expansion stops when the controller judges the collected evidence sufficient, no edge is selected or available, or the depth limit is reached, subject to the recovery rule below. A path-selection call then evaluates each seed's explored paths for useful and complementary evidence. Selected paths receive model-assigned values and enter the shared packing procedure.

In the reported MQuAKE Full configuration, an insufficient navigation decision with no selected edge triggers a within-graph recovery step when candidate edges remain: the neighbour with the highest frozen semantic score is selected.

Selection budgets and context construction. Both panels impose a final budget of at most six distinct memory nodes and 2,000 context tokens, measured with the `o200k_base` tokenizer. Graph-based methods retain at most three paths in total, not three paths per seed in the final answer context. Full, Untyped and w/o Applicability permit at most three graph transitions per path in the conversational panel and four in the MQuAKE panel. Semantic Retrieval does not construct paths.

For graph-based methods, candidate paths are processed in descending score order, with ties resolved by shorter path length and then fixed identifier order. Paths that add no new nodes are skipped. A path is retained only if its addition respects the path, unique-node, and context-token limits. Paths and memory text are not truncated to force them into the budget.

The answer context contains deduplicated memory summaries, ordered by stored session date with memory identifier as a tie-breaker. Within each panel, all arms use this same renderer. Edge labels, path scores, and reference annotations are not supplied to the answer model.

Answer models and realized computation. Within each panel, the arms share the answer prompt, answer model, and judging configuration. The conversational panel uses GPT-4o-mini for answer generation and GPT-4.1-mini for judging. The MQuAKE-derived panel uses GPT-4.1-mini for both. Each reported configuration has one recorded execution per question.

Conversational metrics. Let M_q be the final selected memory set, G_q the normalized official reference-turn set, and $U(m)$ the source-turn identifiers attached to memory m. Provenance Recall is

$$\mathrm{PR}(q) = \frac{\left|G_q \cap \bigcup_{m \in M_q} U(m)\right|}{|G_q|}.$$

It measures reference-provenance coverage, not semantic entailment of every source fact by its stored summary.Applicable Precision is the fraction of selected memories labelled required, sufficient, or helpful.

MQuAKE metrics. *Complete Valid Trace* is a question-level binary metric. It is one if at least one retained path exactly matches the designated reference-node sequence, and zero otherwise. The score is macro-averaged over all 400 questions.

For question q, let G_q denote its reference edges and S_q the edge occurrences traversed by its retained paths. *Reference-edge Recall* is

$$R_{\mathrm{edge}}(q) = \frac{1}{|G_q|} \sum_{g \in G_q} \mathbb{I}[\exists e \in S_q : \mathrm{ends}(e) = \mathrm{ends}(g) \wedge r(e) = r(g)],$$

where endpoint matching is unordered and $r(\cdot)$ is the graph-supplied native MQuAKE predicate. Each reference edge can contribute at most once, even if it appears in multiple retained paths.

Invalid Edge Rate is defined operationally as

$$I_{\mathrm{edge}}(q) = \frac{\sum_{e \in S_q} \mathbb{I}[e \notin G_q]}{|S_q|}.$$

It therefore measures the proportion of retained edge occurrences that fall outside the designated reference chain. Questions without retained edges have no defined edge-rate denominator and are excluded from this rate; the corresponding evaluable count is reported with the result. The term "invalid" refers only to disagreement with the constructed reference chain and does not imply that an off-reference edge is semantically or logically incorrect.

These metrics evaluate recovery of dataset-derived reference chains and edges. Because relation labels are supplied as graph metadata, they should not be interpreted as independent relation-type predictions made by the controller.

Table 12 reports the hop-stratified results. The largest descriptive difference occurs in the three-hop subset; two-hop answer accuracies are nearly equal.

B.6 Complete hyperparameter configuration

The confirmed default configuration for applicability tracing is summarized in Table 13. The main sensitivity analysis varies H, B, K_i, and τ_R one at a time around the default configuration; sensitivity analyses for the remaining six hyperparameters are reported in the complete configuration table above.

B.7 Component Ablation Protocol

This section documents the exact intervention applied in each ablation arm. All arms share the frozen full-model configuration.

Table 12: Hop-stratified results (%) on the MQuAKE-T. Answer Accuracy uses FINAL-answer-only judging. Hop counts follow the dataset's number of chain facts.

Hops	n	Complete Valid Trace (%)		Answer Accuracy (%)	
		Full	w/o	Full	w/o
2	303	96.04	96.37	90.76	90.43
3	95	87.37	55.79	86.32	50.53
4	2	100.00	50.00	100.00	50.00
All	400	94.00	86.50	89.75	80.75

Parameter	Function	Default
K_i	Initial seeds per query	5
P	Traces expanded per seed	5
H	Maximum trace depth	3
B	Unique-experience budget	10
τ_R	Relation-confidence threshold	0.95
τ_{fact}	Atomic-reranker relevance score threshold	0.5
K_{fact}	Atomic-evidence budget	10

Table 13: Confirmed default hyperparameters for WeaveMem.

Relational Graph (RG). Online graph traversal is disabled. Seed retrieval, the sufficiency gate, the fallback stage, the per-seed trace cap P, the experience budget B, and atomic evidence recovery are unchanged, and the graph artifact itself is pinned. The arm therefore measures the contribution of online relational expansion rather than of graph construction.

Long-Level Relations (LLR). The typed experience graph is replaced by a graph induced directly between experiences: pairwise relation judgments are made on experience-level metadata under the same candidate generation, the same seven-type inventory, the same reranker relevance score threshold τ_R, and the same candidate budget K_r, and the replacement graph is consumed by the identical online retrieval stack (H=3, P=5, B=10). Long-level relation induction and its projection onto experiences are thereby removed.

Relation Types (RT). The graph is byte-identical to the full model: topology, stored directions, natural-language edge descriptions, and confidence scores are unchanged. Only the discrete seven-type labels are hidden from the navigation and trace-selection prompts. Edge descriptions may still carry inferable relation semantics, so the arm isolates the *explicit* type channel rather than all relation semantics.

Multi-Path Selection (MPS). The per-seed cap on complementary retained traces is reduced P: $5 \to 1$, using the same trace-selection prompt and response schema. The K_i initial seeds, multi-hop traversal within a trace, and the complete-path budget B are unchanged. The arm removes complementary per-seed trace selection, not multi-path retrieval globally.

Evidence Recovery (ER). Answer-time atomic evidence recovery is disabled: the query-relevant atomic memories attached to the finally selected experiences are removed from the answer context.

Guided Fallback (GF). The coverage-recovery stage is disabled: only the initial attempt, seeded by the top K_i experiences of the RRF–rerank stage, is executed. The collective sufficiency gate, the broader fallback pool of C_f candidates, and seed replacement with K_f new seeds never run; trace selection, evidence recovery, answering, and judging are unchanged.

C Additional Results and Analysis

This section reports additional analyses that complement the main benchmark, efficiency, sensitivity, and applicability results.

C.1 Complementary benchmark results

Tables 14 report the complete category-level results for the four benchmarks used in our evaluation: four categories for LoCoMo, six for LongMemEval-S, seven for PersonaMem-32K, and ten for BEAM-1M. For completeness, these tables include categories also presented in the main benchmark table.

LoCoMo and LongMemEval-S use LLM-judge accuracy, PersonaMem-32K uses deterministic exact-choice accuracy, and BEAM-1M reports the mean rubric score within each category. Each entry gives the mean score over five runs, with the sample standard deviation shown in smaller parentheses ($n_{\text{runs}} = 5$, $\text{ddof} = 1$). Means are expressed as percentages and standard deviations as percentage points.

Evidence-turn recall. The evidence-recall columns of Table 15 report native-unit exact-turn Recall-all@K for benchmarks with turn-level evidence annotations. For each question, the top-K retrieved memory units are expanded to their source turns. A question is counted as recalled only if this set covers every annotated evidence turn; covering only some of the required turns does not count as a success. Thus, R@K measures the percentage of questions with complete evidence coverage, rather than the average fraction of evidence turns retrieved.

The recall denominator, n_{R}, includes only questions with non-empty turn-level evidence annotations: 1,536 for LoCoMo, 470 for LongMemEval-S, and 625 for BEAM-1M. PersonaMem-32K is excluded because its released multiple-choice evaluation does not define an analogous turn-level evidence subset.

Because systems retrieve native units with different granularities, these scores measure evidence coverage under each system's own retrieval representation. The same K does not imply the same number of source turns or input tokens. In particular, MemGAS uses complete raw sessions as its native units on LoCoMo and LongMemEval-S, so its recall on these benchmarks reflects session-level evidence coverage. The results should therefore not be interpreted as comparisons under matched passage lengths or token budgets. Dashes in the Full Context row indicate that retrieval recall is not reported, rather than zero recall.

Answer-overlap diagnostics. The answer-overlap columns of Table 15 report token-level F1 and BLEU-1, the two lexical-overlap measures available for every compared system on the three generative benchmarks. LoCoMo and LongMemEval-S are evaluated on the extracted final answer, whereas BEAM-1M uses the complete generated response. PersonaMem-32K is excluded because its official evaluation compares the predicted option with the reference option rather than comparing free-form answer text.

The answer-overlap denominator, n_{A}, covers the complete answer set: 1,540 questions for LoCoMo, 500 for LongMemEval-S, and 700 for BEAM-1M. These counts differ from n_{R} because evidence recall is restricted to questions with non-empty turn-level evidence annotations. All scores are expressed as percentages. F1 and BLEU-1 are descriptive diagnostics computed over one complete answer set per method; run-wise variability, confidence intervals, and paired comparisons were not computed. They measure lexical agreement with the reference answers and should not be treated as direct measures of semantic correctness.

Interpretation of the combined results. On LoCoMo, WeaveMem obtains the highest Recall-all@5 and Recall-all@10, at 83.98% and 90.43%, respectively. These exceed the next-highest scores, both from EverMemOS, by 2.86 and 5.08 percentage points. At $K = 3$, however, EverMemOS leads with 77.02%, compared with 75.98% for WeaveMem. On LongMemEval-S, EverMemOS leads at $K = 3$ (82.98%), while MemGAS leads at $K = 5$ (91.70%) and $K = 10$ (96.17%); WeaveMem reaches 69.79%, 83.83%, and 91.49% at the three cutoffs. On BEAM-1M, EverMemOS leads at all three cutoffs, with 34.56%, 42.88%, and 51.68%, whereas WeaveMem

Table 14: Complete category-level results on LoCoMo, LongMemEval-S, PersonaMem-32K, and BEAM-1M. Each entry reports the mean score (%) over five runs, with the sample standard deviation in percentage points shown in smaller parentheses ($n_{\text{runs}} = 5$, $\text{ddof} = 1$). Bold indicates the highest mean in each category.

LoCoMo

Method	Multi-hop	Temporal	Open-domain	Single-hop
Full Context	88.44 (0.40)	84.42 (1.06)	**76.88** (2.26)	94.67 (0.27)
Mem0	68.44 (1.76)	80.19 (1.11)	62.71 (2.00)	84.71 (0.33)
A-Mem	73.33 (1.16)	77.57 (1.19)	58.54 (2.38)	79.95 (0.39)
Graphiti	61.21 (0.64)	51.53 (0.72)	63.96 (2.28)	62.00 (0.51)
REMem	51.91 (0.74)	82.31 (0.26)	53.13 (1.80)	78.67 (0.14)
Memora	**88.79** (0.96)	88.91 (1.22)	72.29 (0.93)	94.77 (0.44)
MemGAS	44.40 (1.26)	39.75 (1.69)	55.21 (1.65)	73.10 (0.26)
HyperMem	86.45 (1.08)	86.17 (1.70)	68.13 (0.93)	93.96 (0.40)
EverMemOS	86.24 (1.10)	90.78 (0.47)	72.08 (0.87)	93.32 (0.23)
MAGMA	69.01 (0.89)	79.69 (0.89)	66.67 (0.00)	86.59 (0.16)
WeaveMem	88.09 (1.05)	**91.96** (1.02)	73.75 (1.54)	**95.93** (0.46)

LongMemEval-S

Method	Know.	Multi.	Asst.	Pref.	User	Temp.
Full Context	81.79 (3.06)	68.12 (1.47)	97.86 (1.49)	79.33 (6.41)	94.00 (1.86)	56.54 (3.12)
Mem0	88.72 (2.11)	66.32 (0.82)	86.07 (1.49)	93.33 (2.36)	94.86 (1.63)	79.55 (0.98)
A-Mem	87.44 (1.40)	77.59 (3.07)	95.36 (1.60)	84.67 (1.83)	95.43 (0.64)	78.05 (1.24)
Graphiti	66.41 (2.11)	39.10 (1.76)	72.50 (0.98)	93.33 (2.36)	83.43 (0.78)	57.59 (1.65)
REMem	90.51 (0.70)	74.74 (0.67)	78.57 (2.53)	74.67 (4.47)	92.57 (0.64)	76.54 (0.98)
Memora	89.49 (1.07)	**82.11** (1.87)	65.36 (0.98)	94.67 (2.98)	98.00 (1.63)	**89.02** (0.67)
MemGAS	74.36 (0.91)	53.38 (1.19)	98.21 (0.00)	83.33 (4.71)	94.86 (0.78)	63.76 (1.63)
HyperMem	91.54 (1.15)	78.35 (0.82)	91.43 (1.49)	**98.00** (1.83)	95.14 (0.78)	85.11 (1.24)
EverMemOS	83.59 (1.67)	77.44 (1.76)	90.71 (1.49)	96.00 (3.65)	98.29 (0.64)	88.27 (1.14)
MAGMA	89.23 (2.33)	60.30 (2.46)	95.36 (0.98)	88.00 (1.83)	92.57 (1.56)	78.95 (0.53)
WeaveMem	**92.05** (1.40)	80.15 (1.47)	**98.93** (0.98)	97.33 (2.79)	**99.43** (0.78)	84.51 (1.14)

PersonaMem-32K

Method	Gen.	Rec.	Shared	Mentioned	Reason	Ideas	Evol.
Full Context	80.70 (1.75)	72.73 (2.23)	63.41 (1.15)	70.59 (0.00)	**91.92** (1.24)	25.81 (1.70)	71.08 (1.86)
Mem0	71.93 (0.00)	76.73 (1.52)	70.08 (0.69)	64.71 (0.00)	86.06 (0.85)	46.67 (0.96)	67.05 (0.32)
A-Mem	85.26 (0.96)	78.55 (1.99)	**71.32** (1.55)	76.47 (0.00)	87.07 (0.85)	33.55 (1.18)	72.37 (1.09)
Graphiti	72.28 (0.78)	69.82 (1.00)	43.10 (0.69)	56.47 (3.22)	79.60 (1.81)	30.32 (1.18)	58.56 (0.64)
REMem	61.05 (2.88)	74.18 (0.81)	65.27 (0.65)	64.71 (0.00)	81.21 (1.15)	46.24 (1.08)	54.68 (0.51)
Memora	74.74 (1.57)	73.82 (2.07)	65.12 (1.34)	75.29 (2.63)	86.67 (0.45)	**51.61** (1.08)	73.38 (0.72)
MemGAS	59.30 (2.29)	58.18 (1.82)	61.24 (2.12)	77.65 (4.92)	72.12 (0.90)	44.73 (3.45)	52.81 (2.07)
HyperMem	73.68 (1.24)	72.36 (1.99)	70.70 (0.85)	71.76 (4.92)	90.71 (0.45)	44.52 (0.96)	**73.67** (1.66)
EverMemOS	86.32 (0.78)	81.82 (2.23)	65.74 (0.35)	76.47 (0.00)	85.86 (1.24)	37.42 (0.90)	68.78 (0.82)
WeaveMem	**87.72** (2.15)	**85.45** (1.82)	66.20 (2.02)	**88.24** (0.00)	88.08 (1.11)	50.32 (0.48)	70.36 (1.07)

BEAM-1M

Method	Pref.	Instr.	Summ.	Multi.	Abst.	Extr.	Contr.	Know.	Temp.	Order
Full Context	80.43 (1.54)	51.25 (0.97)	68.62 (1.13)	60.22 (1.56)	**82.86** (1.75)	68.33 (0.99)	42.93 (1.35)	9.07 (0.78)	18.33 (1.01)	35.35 (1.43)
Mem0	86.63 (0.87)	79.42 (0.81)	56.68 (0.69)	66.34 (0.88)	67.71 (1.63)	72.54 (1.13)	66.86 (0.71)	72.57 (1.11)	34.14 (0.82)	39.43 (1.58)
A-Mem	88.56 (1.34)	86.39 (1.54)	68.90 (1.15)	74.54 (1.41)	60.86 (2.34)	72.53 (1.71)	54.50 (1.67)	66.14 (1.54)	48.79 (2.17)	34.79 (0.78)
Graphiti	80.79 (1.30)	68.62 (1.75)	69.68 (0.83)	60.32 (0.88)	73.07 (1.98)	64.71 (0.87)	53.50 (0.51)	39.43 (0.82)	38.71 (0.96)	36.55 (0.62)
REMem	85.04 (1.39)	78.17 (1.70)	48.71 (0.73)	65.71 (0.85)	69.93 (3.10)	72.41 (1.15)	70.18 (1.11)	66.50 (1.52)	46.67 (1.52)	43.90 (0.77)
Memora	91.44 (1.21)	**90.46** (1.43)	84.37 (0.42)	72.19 (1.37)	65.57 (3.06)	78.32 (1.36)	65.32 (1.11)	67.36 (2.00)	**57.14** (2.93)	**64.25** (1.75)
MemGAS	85.79 (0.91)	75.77 (0.89)	64.65 (0.31)	62.17 (0.55)	63.57 (1.13)	59.28 (0.65)	45.86 (0.51)	43.36 (0.60)	31.19 (1.31)	28.09 (0.84)
HyperMem	90.83 (1.34)	88.86 (1.64)	79.10 (0.70)	**78.66** (1.52)	59.57 (2.45)	75.78 (1.64)	61.36 (0.39)	**72.71** (2.27)	47.02 (2.15)	52.39 (1.53)
EverMemOS	90.30 (0.83)	89.58 (1.04)	**86.67** (0.42)	74.05 (1.13)	62.79 (1.81)	77.24 (1.31)	57.93 (0.92)	62.00 (2.12)	53.14 (1.35)	57.28 (0.86)
WeaveMem	**91.60** (0.38)	87.48 (0.77)	79.09 (0.61)	76.75 (0.46)	62.14 (3.07)	**93.44** (0.94)	**73.75** (1.26)	67.43 (1.22)	56.71 (1.27)	52.77 (2.75)

Table 15: Evidence-turn recall and lexical answer-overlap diagnostics. R@K denotes native-unit exact-turn Recall-all@K.

	Evidence-turn recall (%)				Answer overlap (%)		
Method	R@3	R@5	R@10	n_{R}	F1	BLEU-1	n_{A}
LoCoMo							
Full Context	—	—	—	—	24.70	14.51	1,540
Mem0	50.59	56.51	62.30	1,536	21.08	12.23	1,540
A-Mem	28.06	33.92	43.16	1,536	19.13	11.00	1,540
Graphiti	28.45	30.21	31.97	1,536	17.15	10.09	1,540
REMem	70.77	75.85	78.97	1,536	**25.10**	**16.48**	1,540
Memora	69.66	75.65	81.25	1,536	21.95	12.26	1,540
MemGAS	54.95	66.93	81.18	1,536	20.73	13.00	1,540
HyperMem	66.08	72.53	79.62	1,536	20.93	11.64	1,540
EverMemOS	**77.02**	81.12	85.35	1,536	20.86	12.05	1,540
WeaveMem	75.98	**83.98**	**90.43**	1,536	22.44	12.70	1,540
LongMemEval-S							
Full Context	—	—	—	—	22.02	12.16	500
Mem0	66.81	74.68	83.40	470	22.54	14.38	500
A-Mem	31.49	45.74	64.26	470	22.85	14.48	500
Graphiti	25.32	29.15	33.62	470	20.26	13.11	500
REMem	71.70	84.04	91.49	470	**26.46**	**17.82**	500
Memora	76.17	83.62	89.79	470	22.88	14.53	500
MemGAS	81.91	**91.70**	**96.17**	470	24.33	16.15	500
HyperMem	63.62	78.30	89.57	470	22.16	13.70	500
EverMemOS	**82.98**	89.79	95.53	470	22.71	14.56	500
WeaveMem	69.79	83.83	91.49	470	22.68	14.10	500
BEAM-1M							
Full Context	—	—	—	—	30.00	17.73	700
Mem0	13.12	18.24	24.16	625	27.17	13.45	700
A-Mem	23.84	31.20	42.56	625	24.44	10.54	700
Graphiti	8.16	10.24	13.12	625	27.59	14.41	700
REMem	21.92	30.88	38.72	625	29.53	14.83	700
Memora	30.56	40.64	50.56	625	25.35	11.04	700
MemGAS	11.52	19.20	31.52	625	**30.98**	**18.92**	700
HyperMem	13.92	19.20	29.44	625	22.86	8.71	700
EverMemOS	**34.56**	**42.88**	**51.68**	625	23.65	9.94	700
WeaveMem	13.44	21.12	31.20	625	23.81	9.39	700

Table 16: Offline memory-graph construction cost on LoCoMo-10 (10 conversations, 5,882 turns).

	EVERMEMOS	HYPERMEM	WEAVEMEM
Memory store			
Episodes / experiences	452	767	519
Atomic memories / facts	732	6,093	4,290
Higher-order structures	831 clusters	333 topics	—
Relations / hyperedges	—	1,096	6,053
Storage	320.0 MiB	206.3 MiB	166.9 MiB
Build cost			
Build time	0.7 h	4.5 h	2.5 h

obtains 13.44%, 21.12%, and 31.20%. WeaveMem's stronger evidence coverage on LoCoMo at the larger cutoffs therefore does not extend uniformly across the other benchmarks.

The lexical-overlap rankings also differ from the evidence-recall rankings. REMem achieves the highest F1 and BLEU-1 on LoCoMo (25.10% and 16.48%) and LongMemEval-S (26.46% and 17.82%), while MemGAS leads on BEAM-1M (30.98% and 18.92%). WeaveMem's corresponding F1/BLEU-1 scores are 22.44%/12.70%, 22.68%/14.10%, and 23.81%/9.39%. For example, its leading Recall-all@10 on LoCoMo does not coincide with the highest lexical overlap. The combined table thus distinguishes complete evidence coverage from surface-form agreement in generated answers. These aggregate comparisons do not establish a causal relationship between retrieval coverage and answer quality; neither diagnostic replaces the benchmark-specific answer evaluation reported in the main results.

C.2 ADDITIONAL EFFICIENCY RESULTS

Metric and token scope. The plotted quantity is the mean number of answer-model input tokens per query, defined as the API-reported `prompt_tokens` of the first API-successful answer-model request for each question, or the originating request's usage when an exact answer cache is hit. The count includes the full answer prompt—instructions, the user query, and the retrieved memory text or conversation history. A first request need not produce a format-valid answer; accuracy is scored from the original experiment's final answer, which may have followed format retries. The paired metric therefore does not estimate accuracy with retries disabled. For HyperMem only the stage-5 answer call is counted (its ingestion-stage calls are excluded); for EverMemOS we use the per-question answer-input usage reported in its run artifacts.

Evaluated queries and aggregation. Each run contains 500 questions for LongMemEval-S or 589 questions for PersonaMem-32K. We first compute accuracy and mean answer-input tokens over all questions within each run, and then report the arithmetic mean and sample standard deviation across the five repetitions ($n = 5$, $\mathrm{ddof} = 1$).

On LongMemEval-S, accuracy is an LLM-judge verdict: all methods are evaluated by a single `gpt-4.1-mini` judge at temperature 0 under the same protocol. On PersonaMem-32K there is no LLM judge: all methods are scored by the deterministic official set-equality parser over multiple-choice answers, so accuracy figures are directly comparable across methods.

Reading of the figure. WeaveMem does *not* use the fewest input tokens on either benchmark. Mem0, Graphiti, REMem, MemGAS, Memora, and EverMemOS are lower on LongMemEval-S; Mem0, Graphiti, REMem, MemGAS, and EverMemOS are lower on PersonaMem-32K. We therefore focus the efficiency comparison on HyperMem, the highest-accuracy executed baseline on both benchmarks under the five-run mean. Relative to HyperMem, WeaveMem attains higher mean accuracy while using $1.84\times$ and $2.14\times$ fewer answer-model input tokens per query on LongMemEval-S and PersonaMem-32K, respectively.

Figure 6 extends the accuracy–context comparison in the main paper to LongMemEval-S and PersonaMem-32K. On LongMemEval-S, `WeaveMem` obtains the highest overall LLM-judge ac-

Table 17: Five-run mean per-query answer-model input tokens and same-batch overall accuracy.

Benchmark	Method	Accuracy (%)	Mean input tokens (exact)	Plotted
LongMemEval-S ($n = 500$)	WeaveMem	89.0000 ± 0.5831	5,986.73	5,987
	Mem0	81.1600 ± 0.2966	1,681.97	1,682
	A-Mem	84.1600 ± 0.7925	12,063.74	12,064
	Graphiti	61.4800 ± 0.7294	2,796.78	2,797
	REMem	80.6000 ± 0.2400	1,266.05	1,266
	MemGAS	72.0400 ± 0.4336	993.39	993
	HyperMem	87.2000 ± 0.2000	10,996.45	10,996
	Memora	86.2000 ± 0.5831	4,946.03	4,946
	EverMemOS	86.8000 ± 0.2000	5,107.21	5,107
	FullContext	74.8000 ± 1.3266	115,677.30	115,677
PersonaMem-32K ($n = 589$)	WeaveMem	72.8693 ± 0.6168	5,165.02	5,165
	Mem0	68.9983 ± 0.2600	1,285.64	1,286
	A-Mem	70.4244 ± 0.4227	12,392.16	12,392
	Graphiti	56.5705 ± 0.2200	1,948.69	1,949
	REMem	62.8523 ± 0.6619	2,200.92	2,201
	MemGAS	58.4720 ± 0.4427	867.62	868
	HyperMem	71.1036 ± 0.6168	11,061.00	11,061
	Memora	70.5942 ± 0.3479	9,123.00	9,123
	EverMemOS	69.1681 ± 0.5036	3,946.00	3,946
	FullContext	66.8251 ± 0.6192	28,069.69	28,070

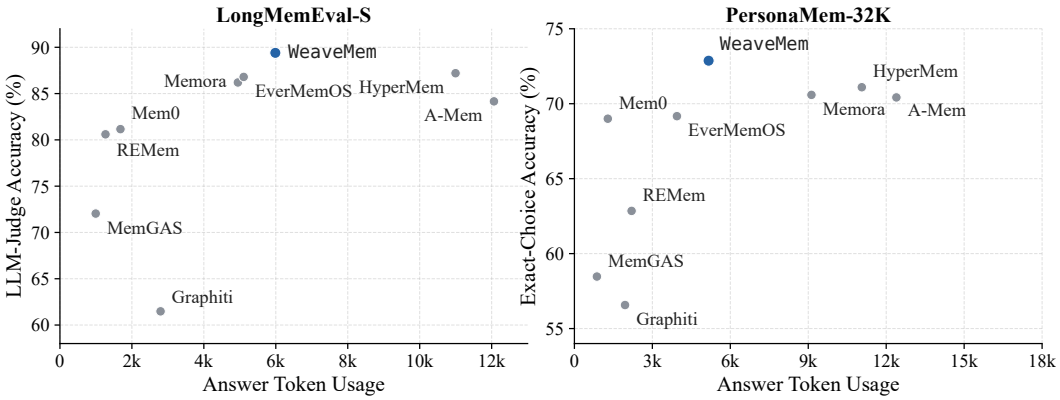


Figure 6: Overall answer accuracy versus average answer-model input tokens on LongMemEval-S and PersonaMem-32K. Upper-left positions indicate higher answer accuracy with a smaller answer context.

curacy with approximately 5.99K answer-model input tokens. This context is substantially smaller than the contexts used by A-Mem and HyperMem, which require approximately 12.06K and 11.00K tokens, respectively. On PersonaMem-32K, `WeaveMem` again obtains the highest overall accuracy with approximately 5.17K tokens, compared with 12.39K for A-Mem, 11.06K for HyperMem, and 9.12K for Memora.

These results complement the LoCoMo and BEAM-1M analyses in the main paper. WeaveMem does not minimize token usage in every comparison. Instead, it achieves the strongest answer quality with a moderate answer context and occupies a favorable accuracy–context region relative to the strongest high-accuracy baselines.

Hop	Complete Valid Trace	Invalid Edge ↓	Answer Acc. (Judge)	Final-answer Alias Match
2-hop	96.0 / 96.4	0.0 / 55.8	90.8 / 90.4	90.4 / 89.8
3-hop	87.4 / 55.8	4.3 / 31.8	86.3 / 50.5	91.6 / 54.7

Table 18: Hop-stratified MQuAKE-T results (%). Each cell reports Full `WeaveMem` / `WeaveMem` w/o Applicability.

C.3 Additional hyperparameter sensitivity

Design. Each panel varies exactly one retrieval hyperparameter; all other components are fixed at the vector that produced the main LoCoMo result: H=3, P=5, B=10, K_i=5, K_{fact}=10, τ_{fact}=0.5, K_f=10, C_i=20, C_f=50, K_r=10, τ_r=0.95, and path-score weights $(w_n, w_t, w_d, w_c) = (0.6, 0.2, 0.1, 0.1)$. The parameters are: P, the maximum number of controller-selected paths per seed; K_{fact}, the conversation-wide ceiling on post-path child-fact selection; τ_{fact}, the child-fact rerank score threshold; K_f, the fallback seed count; C_i and C_f, the initial and fallback RRF candidate pool sizes. Defaults were fixed *a priori* from the frozen configuration, not from the best-performing point of any sweep.

Evaluation protocol. Every arm answers and is judged on all 1,540 scored LoCoMo questions (282 multi-hop, 321 temporal, 96 open-domain, 841 single-hop; the adversarial category is excluded by the benchmark protocol). Each arm is a single run; the default point is evaluated once and reused within each panel.

Reading of the K_{fact} panel. All six non-default arms lie within ± 16 questions (± 1.04pp) of the K_{fact}=10, comparable to the run-to-run noise floor of this evaluation, and the curve is non-monotone. The nominal fresh-grid best point (K_{fact}=100, 1432/1540) exceeds the reference by a single question and is not supported as an improvement; the default K_{fact}=10 is retained.

Figure 7 reports sensitivity to six additional hyperparameters on LoCoMo. Each panel varies one parameter while keeping the remaining configuration fixed. Results are shown for single-hop, multi-hop, temporal, and open-domain questions.

The results show comparatively stable single-hop performance across the tested settings. Multi-hop and temporal performance vary more because these categories depend on trace depth, candidate coverage, and evidence selection. Open-domain questions exhibit larger fluctuations, reflecting their more diverse evidence requirements.

Increasing the number of retained paths per seed P does not produce a consistent gain beyond a moderate value. Increasing the atomic-evidence budget K_{fact} also provides limited additional benefit after the relevant evidence is covered. Performance remains broadly stable across the tested values of the atomic-evidence threshold τ_{fact}. The fallback seed count K_f and candidate-pool sizes C_i and C_f mainly control recovery when the initial retrieval does not expose sufficient evidence. Across the tested ranges, the model does not depend on a narrowly tuned setting for these parameters.

C.4 Complementary evaluation results

To complement the aggregate MQuAKE-T results, we stratify relation-aware trace recovery by the number of required hops. This analysis tests whether the benefit of applicability-guided tracing persists as the evidence chain becomes longer.

The three-hop subset provides the most informative comparison for multi-step composition. Full `WeaveMem` recovers 87.4% complete valid traces compared with 55.8% for the ablation, reduces the invalid edge rate from 31.8% to 4.3%, and improves judge-based answer accuracy from 50.5% to 86.3%. On two-hop questions the ablation is descriptively slightly higher on complete valid traces (96.4% vs. 96.0%) because it retains about three paths per question instead of one, yet this does not translate into an answer-accuracy gain—the two arms are within a single question (90.8% vs. 90.4%)—while the same multi-path behavior floods its context with Invalid Edge(55.8% vs. 0.0% invalid edge rate), so we do not claim a uniform per-hop coverage advantage. The four-hop subset

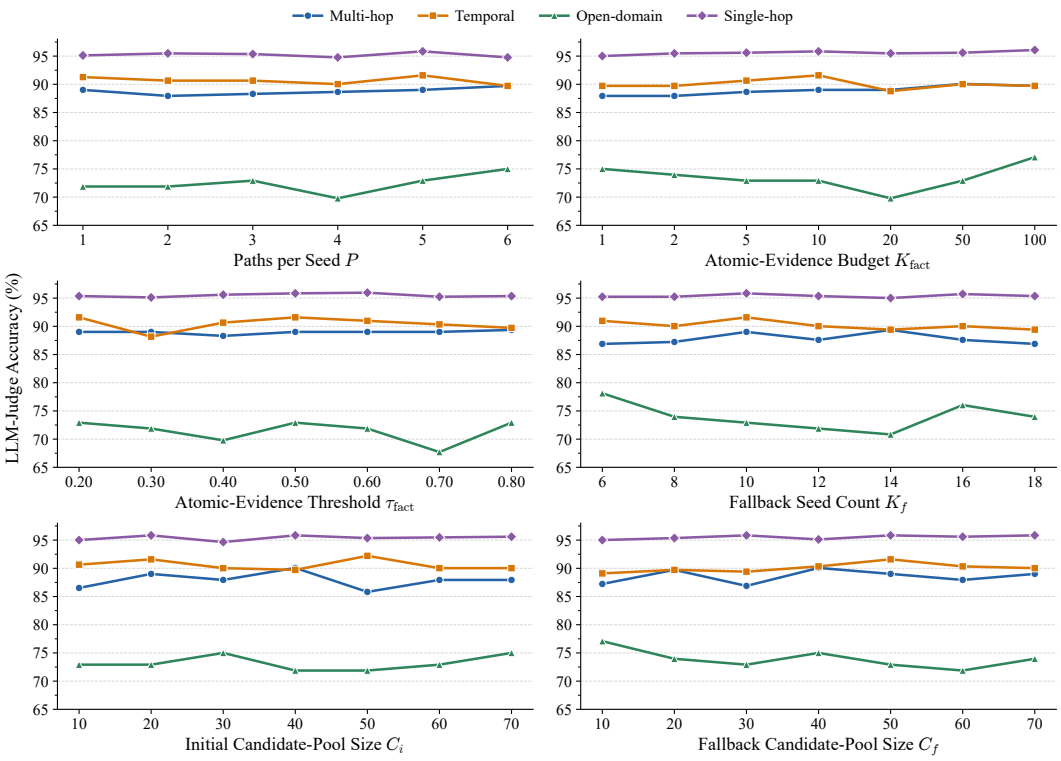


Figure 7: Additional hyperparameter sensitivity on LoCoMo. From left to right and top to bottom, the panels vary the paths retained per seed P, atomic-evidence budget K_{fact}, atomic-evidence threshold τ_{fact}, fallback seed count K_f, initial candidate-pool size C_i, and fallback candidate-pool size C_f. Each panel varies one parameter while keeping all other settings fixed.

contains only two examples—the entire four-hop population of MQuAKE-T—and is therefore not used for a quantitative conclusion.

C.5 Qualitative evidence traces

The qualitative analysis illustrates the recorded evidence-selection decisions behind the aggregate results. We present two post-hoc examples covering fact-chain recovery, retrieval of a dated session memory, and an empty-context outcome with a disputed reference answer.

Within each specified behavior group, we use the first eligible case in frozen result-file order: `mq1432` among the 33 three-hop cases where Full `WeaveMem` retains a complete reference chain, the ablation does not, and Full `WeaveMem` is judged correct; `conv-26:45` among the 20 correctly answered Full `WeaveMem` cases whose first retained edge is labeled UPDATES. This deterministic within-group rule makes selection reproducible, but does not establish statistical representativeness or eliminate post-hoc selection bias.

C.6 Failure analysis

Full `WeaveMem` makes 87 errors on the 252-question conversational panel (65.5% accuracy) and 41 errors on the 400-question MQuAKE-T panel (89.75% accuracy), yielding 128 errors in total. Because the two panels represent evidence differently, we analyze the observable evidence available to the answer generator rather than forcing all errors into a shared, fine-grained causal taxonomy.

For the conversational panel, each context is assessed twice and disagreements are adjudicated; empty contexts are deterministically marked insufficient. For MQuAKE-T, we programmatically test whether the final context contains every node in the dataset-derived reference chain. These def-

Table 19: Case 1: three-hop fact-chain recovery on MQuAKE-T (`mq1432`).

Query	What is the name of the leader of the country where Aki Takase's music genre originated?
Seed memory	The record `1432:f0`, "Aki Takase genre jazz.", serves as the chain-start seed. Synthetic ordering timestamps are omitted.
Recorded expansion	At the first expansion, the navigator selects gold edge `1432:e0` and does not select distractor edge `1432:dx2`. At the second expansion, it selects gold edge `1432:e1` together with distractor edges `1432:dx3` and `1432:dx4`. Expansion then stops because no unvisited edges remain.
Retained trace	`seed1-path6`, controller value 1.0. The controller also selects the shorter prefix `seed1-path3` with value 0.8; whole-path packing skips that prefix because it contributes no new nodes. The final sufficiency check returns *sufficient*.
Intermediate memories	`1432:f1`: "jazz country of origin United States of America."; `1432:f2`: "United States of America head of government Donald Trump."
Answer-bearing evidence	The three fact sentences form the final 27-token context. None of the 20 distractor fact nodes is retained in that context, although two distractor branches are explored during navigation.
Evidence context	"Aki Takase genre jazz. / jazz country of origin United States of America. / United States of America head of government Donald Trump."
Answer / judge	"Donald Trump" — CORRECT (reference: "Donald Trump"). The ablation retains the singleton `1432:f0`, the one-edge path `1432:f0`→`1432:f1`, and the distractor path `1432:f0`→`1432:d2` ("Don Lusher genre jazz."). None contains the head-of-government fact. Its answer names Joe Biden, which is absent from its retrieved context, and is graded WRONG.
Interpretation	This example illustrates complete-chain retention after exploration and path selection, rather than perfect rejection of distractors at every expansion. The benchmark's three-hop chain comprises three fact nodes and two inter-fact transitions.

initions are panel-specific: "sufficient" denotes semantic answerability in the conversational panel, whereas "complete" denotes exact reference-chain coverage in MQuAKE-T. We therefore report them as evidence-status measurements rather than interchangeable causal labels.

Missing or insufficient evidence is the largest observable error state, accounting for 71 of 128 errors (55.5%). On MQuAKE-T, the 15 incomplete-chain errors can be localized further: in eight cases, expansion did not traverse the complete reference chain; in the remaining seven, the reference edges were traversed, but the final path-selection step did not retain the complete chain in the packed context. This distinction suggests that both broader or backtracking expansion and stronger final path retention could improve multi-hop coverage.

Finally, 18 errors terminate with no selected evidence: 15 in the conversational panel and three in MQuAKE-T. All 18 receive incorrect final answers. Trace inspection nevertheless shows that empty selection can arise through different control paths, so it should not be interpreted uniformly as expansion failing to start. An empty-context rescue mechanism may reduce this failure mode, but it cannot address incomplete non-empty contexts, path-retention errors, or evaluation-side discrepancies.

Overall, the audit identifies insufficient final evidence as the largest observable error state, but it does not support attributing a majority of errors to one retrieval subsystem. The MQuAKE-T traces specifically motivate improvements to both graph expansion and final path retention, while the complete-context cases indicate that answer evaluation and reference quality must be audited separately from retrieval quality.

D PROMPTS

This appendix records the prompts used by the LLM-based components of `WeaveMem`. Where a system message is used, it is shown separately from the user message; otherwise, the call consists of a single user message. Runtime placeholders are shown in braces and defined with each prompt. The displayed prompt text matches the final implementation.

Table 20: Case 2: retrieval of a dated session memory (`conv-26:45`). A path containing reverse UPDATES and forward EXPLAINS traversal reaches the August record. The absolute answer date is already explicit in the stored summary, and the ablation retrieves identical answer context.

Query	When did Caroline attend a pride parade in August?
Seed memories	`160496ff`, recorded on 2023-07-03, and `3ea50c2d`, recorded on 2023-07-15, describe earlier parade attendance but do not contain the requested August date. These are session-record dates, not independently verified dates of the parade.
Retained trace	`seed2-path114`: `3ea50c2d` $\xrightarrow{\text{UPDATES (reverse)}}$ `160496ff` $\xrightarrow{\text{EXPLAINS (forward)}}$ `35ee5c23`, controller value 1.0. The stored UPDATES edge `bdfb0437` points from `160496ff` to `3ea50c2d` and is traversed in reverse. The stored EXPLAINS edge `e6d0946f` points from `160496ff` to `35ee5c23` and is traversed forward.
Endpoint memory	The record `35ee5c23`, dated 2023-08-14, states that Caroline "attended a pride parade last Friday (i.e., 2023-08-11)."
Answer-bearing excerpt	The explicit parenthetical date in the August 14 stored summary: "last Friday (i.e., 2023-08-11)." The underlying dialogue says "last Friday"; the absolute date has already been inserted into the memory presented to the answer model.
Evidence context	Both Full `WeaveMem` and the ablation receive the same three session summaries, ordered July 3, July 15, and August 14. Their rendered contexts and answer prompts are byte-identical.
Answer / judge	"Caroline attended a pride parade on Friday, August 11, 2023." — CORRECT (reference: "The Friday before 14 August 2023"). The ablation produces the same answer from the same context and is also graded CORRECT.
Interpretation	The recorded path reaches an answer-bearing endpoint absent from the seeds. The relation names and directions describe the stored graph; they do not independently establish a causal or temporal relation between the earlier and August events. Because the endpoint already contains the absolute date and the ablation retrieves identical context, this case demonstrates a successful retrieval route, not the necessity of that relation sequence or independent answer-time date resolution.

Evidence status	**Conversational**	**MQuAKE-T**	**Total**
Incomplete or insufficient	56 (64.4%)	15 (36.6%)	71 (55.5%)
Complete or sufficient	27 (31.0%)	26 (63.4%)	53 (41.4%)
Sufficiency unresolved	4 (4.6%)	0 (0.0%)	4 (3.1%)
All errors	87	41	128

Table 21: Evidence status for all errors made by Full `WeaveMem`.

D.1 QUERY ANALYSIS AND RELATION PRIOR

Before tracing, `WeaveMem` analyzes the query with a deterministic rule-based classifier; this step makes no LLM call and therefore has no prompt. Regular expressions over the lowercased query assign one of nine question types, evaluated in the priority order `when`, `how`, `why`, `where`, `who`, `which`, `what`, `yes_no`, `other`. The `when` pattern also covers `what date/time/day/month/year/age`, `how long/old/soon/recently`, and the words `before`, `after`, `earlier`, `later`; `yes_no` requires the query to begin with an auxiliary verb (`did`, `does`, `do`, `is`, `are`, `was`, `were`, `has`, `have`, `had`, `can`, `could`, `will`, `would`). The analysis returns a fixed four-field strategy object: `question_type`, `preferred_relation_types`, `direction_guidance`, and `policy`, the latter always the string `soft_type_prior_with_description_override`. The implementation extracts no entity, event, attribute, or time fields from the query. Information requirements are instead represented dynamically during traversal as the controller's free-text `missing` field (Appendix D.2), not as structured slots.

The relation prior is an unweighted set of relation-type names per question type (Table 23), serialized as a JSON array inside the strategy object. It carries no scores, ordering, or probabilities; the

Case	Auditable observation	Interpretation
mq1519	The question asks for the head of government in the country where Louis Banks's genre originated. The final context contains the facts that Louis Banks is associated with jazz and that jazz originated in the United States, but it omits the leader fact. The generated answer identifies only the office—the President of the United States—rather than the reference person, Donald Trump.	An incomplete reference chain prevents the generator from recovering the requested entity. This is a direct example of missing final evidence.
conv-26:27	The system returns an empty selected context. However, the reference claim that Melanie would probably not pursue writing is itself unsupported: the source establishes that she is considering counseling or mental-health work and that she enjoys reading, but it does not establish whether she will pursue writing.	The empty context is a retrieval failure under the evaluation protocol, but the case is not clean evidence of a uniquely attributable system failure because the reference answer is ambiguous.
conv-41:63	The selected evidence places Coco's arrival around July 28, 2023 and Shadow's adoption around August 6, 2023. The resulting interval is approximately nine days, consistent with the generated estimate of about one week and two days, whereas the reference answer states two weeks.	The generated answer follows the available dates more closely than the reference. Counting this case as an answer-generation failure would therefore be misleading.
mq1235	The generated answer states that the head of state of Estonia was "Kersti Kaljulaid," exactly identifying the reference entity, but the automated evaluator marks the answer incorrect.	This is a clear evaluator false negative rather than a retrieval or generation failure.

Table 22: Auditable examples illustrating incomplete evidence, reference ambiguity, reference contradiction, and evaluator error. These examples diagnose observable outcomes without assigning unsupported fine-grained causal labels.

type `other` maps to all seven relation types. The prior is soft guidance in all three places where it is consumed, and no code path ever filters candidate edges by type. First, the traversal-controller prompt states verbatim that "the preferred relation types are a soft prior, not a hard allowlist." Second, the path-selection prompt repeats the same rule ("only as a soft prior. Concrete descriptions and endpoint facts decide relevance."). Third, the deterministic complete-path budget scorer computes `question_type_fit` as the fraction of a path's steps whose relation type lies in the preferred set (a zero-hop path receives the neutral value 0.5) and adds it to the composite path score with weight $w_t = 0.2$, alongside node relevance (0.6), edge-description support (0.1), and the controller's own path value (0.1).

The strategy object is serialized as JSON and supplied to the trace controllers documented in Appendix D.2. No benchmark-specific query-analysis instructions are used.

D.2 Trace expansion, pruning, and sufficiency evaluation

Trace control uses query-conditioned relation metadata to construct evidence paths rather than expanding every structurally connected memory. The controller receives the current trace and the available frontier information. Unvisited node content remains unavailable until the corresponding edge is selected: an edge card exposes only the relation type, the traversal direction, the stored directed description, and a flag marking whether the next node already belongs to this attempt's seed set.

Four control decisions share the three prompts below, and the fourth uses no model at all. (i) Frontier-edge selection is one call per active seed branch per hop, issued in parallel across branches. (ii) Per-seed trace pruning is one path-selection call per seed branch after expansion ends, followed by a deterministic complete-path budget step that has no prompt. (iii) Collective sufficiency eval-

Table 23: The deterministic relation prior: preferred relation types and the verbatim `direction_guidance` string per question type, in classifier priority order. This table is the complete content of the prior; it is produced by rules, not by a model.

Question type	Preferred relation types	Direction guidance (verbatim)
`when`	precedes, updates, causes	Use precedes/updates to reconstruct time order. Traverse reverse to find what came before a known event and forward to find what followed; descriptions may override the type prior when they contain the missing date or temporal clue.
`how`	causes, enables, explains, precedes	Seek mechanism or process: reverse causes/explains/enables from an outcome to its reason or prerequisite, and follow precedes forward through ordered steps.
`why`	explains, causes, enables	Seek reasons: usually traverse causes/explains/enables from result to source, but follow the description whenever it explicitly points to the missing reason.
`where`	supports, updates, precedes	Use place clues in descriptions; relation type is only a soft prior.
`who`	supports, updates, explains	Use endpoint identity clues in descriptions; relation type is only a soft prior.
`which`	supports, contradicts, updates	Use descriptions to distinguish alternatives and verify the requested fact.
`what`	updates, supports, contradicts, precedes	Seek the missing fact or changed state. Prefer updates/supports/contradicts, while allowing any edge whose description concretely predicts the answer.
`yes_no`	supports, contradicts, updates	Seek confirming, conflicting, or updated evidence before deciding.
`other`	all seven types	Use relation descriptions and direction to follow only answer-bearing paths.

Table 24: Trace-control parameters for the LoCoMo configuration used in the main sensitivity analysis. Hard constraints enforced at configuration load are $C_i \geq K_i$, $K_f > K_i$, $C_f \geq K_f$, $H \geq 0$, $P \geq 1$, and $B \geq 1$.

Symbol	Config variable	Default	Meaning
K_i	`GRAPH_INITIAL_SEED_TOP_N`	5	initial seed width
C_i	`GRAPH_INITIAL_RRF_CANDIDATE_TOP_N`	20	initial RRF candidate pool
K_f	`GRAPH_FALLBACK_SEED_TOP_N`	10	fallback seed width
C_f	`GRAPH_FALLBACK_RRF_CANDIDATE_TOP_N`	50	fallback RRF candidate pool
H	`GRAPH_MAX_HOPS`	3	maximum expansion hops per branch
P	`GRAPH_MAX_PATHS_PER_SEED`	5	per-seed selected-path cap
B	`GRAPH_COMPLETE_PATH_MAX_UNIQUE_NODES`	10	global unique-node budget

uation is a single gate call judged over the ordered unique node union of the initial attempt. (iv) Fallback activation is a deterministic rule on the gate verdict, not a model decision. All three calls are single user messages with no system message; the navigation and path-selection models default to the extraction model. Stored direction and traversal direction are shown separately on every edge card, and multiple edges may be selected in one call. Table 24 summarizes the controller parameters used in the LoCoMo configuration.

D.2.1 TRACE EXPANSION

The expansion controller selects frontier edges that can contribute to an uncovered query requirement. It may stop when the current trace is sufficient, no useful edge remains, or the maximum depth is reached.

Trace-expansion prompt

```
You are one of five independent path-search agents navigating a mid-term memory graph. Use
     the question type, stored relation direction, relation type, and concrete edge
     description to select only paths likely to answer the question. Do not answer the
     question itself.

## Question

{{QUESTION}}

## Question-type strategy

{{QUESTION_STRATEGY}}

The preferred relation types are a soft prior, not a hard allowlist. A non-preferred
     relation is useful when its description clearly predicts the missing answer. A
     preferred relation is not useful merely because its label matches the question type.

## Expansion

Current hop: {{HOP}}
Maximum hops: {{MAX_HOPS}}

## Memories already collected by this seed agent

{{CONTEXT}}

## Unvisited edges from this agent's current frontier

{{EDGES}}

{{NEXT_HOP_INSTRUCTION}}

## Rules

1. Set `sufficient=true` only if this agent's current memories contain a specific answer.
     Then return empty `missing` and `selected_edges`.
2. Otherwise state the missing information precisely and select every edge that is
     strongly likely to recover it; multiple edges are allowed.
3. Use direction semantically. For example, reverse `causes` can recover a cause from a
     known result, while forward `causes` can recover an outcome; reverse `precedes` finds
      an earlier stage and forward finds a later stage.
4. For `when` questions, prefer temporal edges only when their direction and description
     lead toward the requested earlier/later event or time clue.
5. For `how` or `why` questions, prefer causal, enabling, explanatory, or ordered-step
     edges only when their descriptions expose the requested mechanism or reason.
6. For `what`, `who`, `where`, and other fact questions, let the concrete description
     override generic type preference.
7. Reject edges that merely share a theme, repeat a habit, or move away from the missing
     information. Relation-label match alone is never enough.
8. Give one concise reason for every selected edge. Return exact edge IDs and no invented
     facts.

Return ONLY this JSON object, without markdown or extra fields:

{"sufficient":false,"missing":"...","selected_edges":[{"edge_id":"edge-id","reason":"Its
     description points from the known event to the missing earlier date."}]}
```

Placeholder instantiation. {{QUESTION}} is the raw question. {{QUESTION_STRATEGY}} is the question-type strategy JSON. {{HOP}} and {{MAX_HOPS}} are the current hop and H. {{CONTEXT}} renders each collected memory as memory_id, topic, summary. {{EDGES}} renders each frontier edge card as relation_type, traversal_direction, description, and next_node_already_seed. {{NEXT_HOP_INSTRUCTION}} is the fixed sentence "Select using only the relation type, traversal direction, and relation description." followed by the seed-awareness sentence "Each edge also marks 'next_node_already_seed'. A true value means the next node is already present in this attempt's global seed set, even if this branch has not visited it. Prefer a relevant false edge when it can add missing evidence, but keep a true edge when it supplies ordering, verification, or a necessary bridge to later evidence."

Output schema, parser, and stopping conditions. The response must be one JSON object with exactly the keys sufficient, missing, selected_edges; sufficient must be a boolean; every edge_id must belong to the displayed candidate set, with an auto-

matic correction applied only when exactly one candidate `midrel-` identifier sits at Hamming distance one from the returned string (the correction is logged); reasons must be non-empty and unique; `sufficient=true` requires empty `missing` and an empty edge list, and `sufficient=false` requires a non-empty `missing`. Responses that violate this contract are rejected. A branch stops when the controller reports sufficiency, when it selects no edge, when no unvisited frontier edge remains, when the selected edges add no new node, or when H is exhausted; the whole round terminates early once no branch makes progress and none remains active. Edges already evaluated by a branch are not offered to it again in later hops.

D.2.2 TRACE PRUNING

For each seed, the trace-pruning stage retains at most P complementary traces and removes redundant, unsupported, or irrelevant paths. The selected traces then compete, together with every reranked seed as a zero-hop candidate, in a deterministic complete-path budget step under the global unique-node cap B; that step is a weighted reranker-based scorer and uses no prompt.

Trace-pruning prompt

```
You are evaluating the paths explored by one mid-memory graph agent. Select zero to {{
    MAX_PATHS}} paths that provide valuable, non-redundant evidence for answering the
    question. Do not answer the question itself.

## Question

{{QUESTION}}

## Question-type strategy

{{QUESTION_STRATEGY}}

## Seed agent

{{SEED_ID}}

## Candidate paths

{{PATHS}}

## Selection rules

1. You may select no path when this seed and every explored path are irrelevant or too
    speculative.
2. Select multiple paths only when they contain complementary answer-bearing evidence,
    such as time plus event, cause plus outcome, or independent supporting details.
3. Do not select a path whose useful nodes and facts are already fully covered by another
    selected path. Prefer the shorter path when evidence is equivalent.
4. Reward relation types and traversal directions applicable to the question type only as
    a soft prior. Concrete descriptions and endpoint facts decide relevance.
5. Penalize topical detours, repeated themes, speculative links, and hops that add no
    useful evidence.
6. The zero-hop seed path is valid, but do not select it separately when a selected
    expanded path already contains the same seed and all its useful evidence.
7. Each `value` must be from 0 to 1 and measure that path's marginal usefulness after
    considering the other selected paths.
8. Return at most {{MAX_PATHS}} unique provided path IDs. Use an empty list when none is
    valuable.

Return ONLY this JSON object, without markdown or extra fields:

{"selected_paths":[{"path_id":"seed1-path2","value":0.86,"reason":"Adds the requested date
    ."},{"path_id":"seed1-path4","value":0.74,"reason":"Adds the cause needed to connect
    the events."}],"reason":"The two paths provide complementary temporal and causal
    evidence."}
```

Placeholder instantiation and parser. {{SEED_ID}} is the branch's seed memory identifier; {{MAX_PATHS}} is P; {{PATHS}} enumerates the branch's simple paths, including the zero-hop seed path, with per-node `id`, `topic`, `summary`, and `already_seed` flags, and per-step `relation_type`, traversal direction, and `description`. The response must have exactly the keys `selected_paths` and `reason`; each selection must reference a unique displayed `path_id` with a `value` in $[0, 1]$ and a non-empty reason; at most P paths may be returned.

D.2.3 SUFFICIENCY EVALUATION

The sufficiency controller evaluates whether the retained traces and zero-hop candidates collectively cover the query. If the evidence remains insufficient, the implementation deterministically invokes exactly one fallback retrieval and tracing attempt at the wider configuration (K_f, C_f), whose result replaces the initial attempt wholesale; seeds are not individually replaced or merged, and no second fallback exists.

Sufficiency-evaluation prompt

```
Judge whether the retrieved mid-term memories contain enough information to answer the
     question fully and specifically. This is only a retrieval gate; do not answer the
     question.

## Question

{{QUESTION}}

## Retrieved memories

{{CONTEXT}}

## Rules

1. Judge only from the displayed memories. Do not assume hidden child memories or graph
     neighbours.
2. Set `sufficient=true` only when the memories contain a specific answer to every part of
      the question.
3. If any requested fact, time, entity, reason, or step is absent or ambiguous, set `
     sufficient=false` and state exactly what is missing.
4. When sufficient, `missing` must be empty. When insufficient, `missing` must be a
     concise non-empty string.

Return ONLY this JSON object, without markdown or extra fields:

{"sufficient":false,"missing":"the specific missing information"}
```

D.3 ANSWER GENERATION AND LLM-JUDGE EVALUATION

Within each benchmark, all evaluated memory systems use the same GPT-4o-mini answer model and the same benchmark-specific answer prompt. The prompts differ across benchmarks because their questions and official evaluation formats differ. The answer input is constructed solely from the benchmark name, public question fields, and retrieved memories, so the memory system changes only the retrieved context. Generated answers are evaluated by GPT-4.1-mini, except on PersonaMem, which follows its official deterministic option-set protocol and uses no LLM judge.

D.3.1 ANSWER GENERATION

No answer system message exists for any benchmark: every answer call is a single user message. Answer prompts differ across benchmarks; the four released user messages are reproduced below. The memory context is serialized per benchmark as follows. *LoCoMo*: a flat list, one prefixed memory per block, each holding the memory content with its source evidence. *LongMemEval*: each memory is headed by `[Memory i | id | session_id | session_date | chat_ids | user_id | eventTime | mentionTime | originalTimeExpression | timePrecision]` (only non-empty fields appear), followed by its content with child facts embedded; an adapter-supplied recent conversation context and the question date are appended. *PersonaMem*: memories are re-sorted by their synthetic source-mention span and each block is labeled with that span, followed by content, span-labeled child facts, and source evidence. *BEAM*: memories are shown as synthetic `M1..Mn` blocks with `topic_subject`, `summary`, `dialogue_phase`, `user_attitude`, and typed child facts. Speakers appear only inside memory text; no benchmark shows retrieval scores. Abstention is benchmark-specific: the BEAM prompt mandates an exact abstention string when the specific information is absent and an unconditional conflicting-information statement; LongMemEval handles abstention on the judge side for `_abs` questions, so its answer prompt is uniform; LoCoMo gives no abstention instruction; PersonaMem forces a choice.

Answer-generation user message — LoCoMo

```
You are an intelligent memory assistant tasked with retrieving accurate information from
     memories.

# CONTEXT:
You have access to memories from conversations between two speakers. These memories may
contain dates or time references within their text that are relevant to answering the
     question.

# MEMORY FIELD SEMANTICS:
- `Fact subject` is the entity that the memory fact describes.
- `Memory owner` is the conversation participant whose memory scope contains the record.
- `Context` is the scene or conversational background stored with an episodic memory.
- A raw core-memory `entityName` value of `user` is resolved to its corresponding `user_id
     `
  before the memory is shown here.
- `Fact subject` and `Memory owner` may refer to different people.
- Attribute a fact to `Fact subject`, never to `Memory owner` merely because the record
     belongs
  to that participant's memory scope.
- Use `Memory owner` only as provenance and conversational context.
- If `Fact subject` is `unknown`, do not substitute `Memory owner` as the subject. Infer a
  subject only when the memory content itself provides clear evidence.

# INSTRUCTIONS:
Your goal is to synthesize information from all relevant memories to provide a
     comprehensive and accurate answer.
You MUST follow a structured Chain-of-Thought process to ensure no details are missed.
Actively look for connections between people, places, and events to build a complete
     picture. Synthesize information from different memories to answer the user's question
     .
It is CRITICAL that you move beyond simple fact extraction and perform logical inference.
     When the evidence strongly suggests a connection, you must state that connection. Do
     not dismiss reasonable inferences as "speculation." Your task is to provide the most
     complete answer supported by the available evidence.

# CRITICAL REQUIREMENTS:
1. NEVER omit specific names - use "Amy's colleague Rob" not "a colleague"
2. ALWAYS include exact numbers, amounts, prices, percentages, dates, times
3. PRESERVE frequencies exactly - "every Tuesday and Thursday" not "twice a week"
4. MAINTAIN all proper nouns and entities as they appear

# RESPONSE FORMAT (You MUST follow this structure):

## STEP 1: RELEVANT MEMORIES EXTRACTION
[List each memory that relates to the question. Note any date mentioned inside the memory
     text.]
- Memory 1: [content]
- Memory 2: [content]
...

## STEP 2: KEY INFORMATION IDENTIFICATION
[Extract ALL specific details from the memories]
- Names mentioned: [list all person names, place names, company names]
- Numbers/Quantities: [list all amounts, prices, percentages]
- Dates/Times: [list all temporal information]
- Frequencies: [list any recurring patterns]
- Other entities: [list brands, products, etc.]

## STEP 3: CROSS-MEMORY LINKING
[Identify entities that appear in multiple memories and link related information. Make
     reasonable inferences when entities are strongly connected.]
- Shared entities: [list people, places, events mentioned across different memories]
- Connections found: [e.g., "Memory 1 mentions A moved from hometown $\rightarrow$ Memory
     2 mentions A's hometown is LA $\rightarrow$  Therefore A moved from LA"]
- Inferred facts: [list any facts that require combining information from multiple
     memories]

## STEP 4: TIME REFERENCE CALCULATION
[If applicable, convert relative time references]
- Original reference: [e.g., "last year" from May 2022]
- Calculated actual time: [e.g., "2021"]

## STEP 5: CONTRADICTION CHECK
[If multiple memories contain different information]
- Conflicting information: [describe]
- Resolution: [explain which is most recent/reliable]
```

```
## STEP 6: DETAIL VERIFICATION CHECKLIST
- [ ] All person names included: [list them]
- [ ] All locations included: [list them]
- [ ] All numbers exact: [list them]
- [ ] All frequencies specific: [list them]
- [ ] All dates/times precise: [list them]
- [ ] All proper nouns preserved: [list them]

## STEP 7: ANSWER FORMULATION
[Explain how you're combining the information to answer the question]

## FINAL ANSWER:
Provide the concise answer with ALL specific details preserved. Do not include anything
     else (including but not limited to: headers/prefixes, explanations, reasoning, steps,
      examples, extra commentary, disclaimers, acknowledgments, greetings, bullet points,
     or any formatting markers).

---
## INPUT SCHEMA
The retrieved memories are given as a flat list, one memory per line, each line holding
     that
memory's content. A memory may concern either speaker; names and any relevant dates appear
inline in the text. List order does not imply importance.

## INPUT
Retrieved memories:
{{MEMORIES}}

Question: {{QUESTION}}

Now, follow the Chain-of-Thought process above to answer the question:
```

Answer-generation user message — LongMemEval

```
You are an intelligent memory assistant tasked with retrieving accurate information from
     memories.

# CONTEXT:
You have access to memories from conversations between two speakers. These memories may
contain dates or time references within their text that are relevant to answering the
     question.

# MEMORY FIELD SEMANTICS:
- `entity_name` is the subject described by the memory's content.
- `user_id` is the conversation participant who owns this memory record.
- `Context` is the scene or conversational background stored with an episodic or
  procedural memory.
- A raw core-memory `entityName` value of `user` is resolved to its corresponding `user_id
     `
  and shown in the `entity_name` field before the memory is provided here.
- `entity_name` and `user_id` may refer to different people.
- Attribute a fact to `entity_name`, never to `user_id` merely because that participant
     owns
  the memory record.
- Use `user_id` only as provenance and conversational context.
- If `entity_name` is `unknown`, do not substitute `user_id` as the subject. Infer a
  subject only when the memory content itself provides clear evidence.

# INSTRUCTIONS:
Your goal is to synthesize information from all relevant memories to provide a
     comprehensive and accurate answer.
You MUST follow a structured Chain-of-Thought process to ensure no details are missed.
Actively look for connections between people, places, and events to build a complete
     picture. Synthesize information from different memories to answer the user's question
     .
It is CRITICAL that you move beyond simple fact extraction and perform logical inference.
     When the evidence strongly suggests a connection, you must state that connection. Do
     not dismiss reasonable inferences as "speculation." Your task is to provide the most
     complete answer supported by the available evidence.

# CRITICAL REQUIREMENTS:
1. NEVER omit specific names - use "Participant A's colleague Participant B" not "a
     colleague"
2. ALWAYS include exact numbers, amounts, prices, percentages, dates, times
3. PRESERVE frequencies exactly - "every Tuesday and Thursday" not "twice a week"
4. MAINTAIN all proper nouns and entities as they appear
```

```
# RESPONSE FORMAT (You MUST follow this structure):

## STEP 1: RELEVANT MEMORIES EXTRACTION
[List each memory that relates to the question. Note any date mentioned inside the memory
     text.]
- Memory 1: [content]
- Memory 2: [content]
...

## STEP 2: KEY INFORMATION IDENTIFICATION
[Extract ALL specific details from the memories]
- Names mentioned: [list all person names, place names, company names]
- Numbers/Quantities: [list all amounts, prices, percentages]
- Dates/Times: [list all temporal information]
- Frequencies: [list any recurring patterns]
- Other entities: [list brands, products, etc.]

## STEP 3: CROSS-MEMORY LINKING
[Identify entities that appear in multiple memories and link related information. Make
     reasonable inferences when entities are strongly connected.]
- Shared entities: [list people, places, events mentioned across different memories]
- Connections found: [e.g., "Memory 1 mentions A moved from hometown $\rightarrow$  Memory
      2 mentions A's hometown is City A $\rightarrow$  Therefore A moved from City A"]
- Inferred facts: [list any facts that require combining information from multiple
     memories]

## STEP 4: TIME REFERENCE CALCULATION
[If applicable, convert relative time references]
- Original reference: [e.g., "last year" from May 2022]
- Calculated actual time: [e.g., "2021"]

## STEP 5: CONTRADICTION CHECK
[If multiple memories contain different information]
- Conflicting information: [describe]
- Resolution: [explain which is most recent/reliable]

## STEP 6: DETAIL VERIFICATION CHECKLIST
- [ ] All person names included: [list them]
- [ ] All locations included: [list them]
- [ ] All numbers exact: [list them]
- [ ] All frequencies specific: [list them]
- [ ] All dates/times precise: [list them]
- [ ] All proper nouns preserved: [list them]

## STEP 7: ANSWER FORMULATION
[Explain how you're combining the information to answer the question]

## FINAL ANSWER:
Provide the concise answer with ALL specific details preserved. Do not include anything
     else (including but not limited to: headers/prefixes, explanations, reasoning, steps,
      examples, extra commentary, disclaimers, acknowledgments, greetings, bullet points,
     or any formatting markers).

---
## INPUT SCHEMA
The retrieved memories are given as a flat list, one memory per line, each line holding
     that
memory's content. A memory may concern either speaker; names and any relevant dates appear
inline in the text. List order does not imply importance.

## INPUT
Question Date:{{QuestionDate}}
Retrieved memories:
{{MEMORIES}}

Question: {{QUESTION}}

Now, follow the Chain-of-Thought process above to answer the question:
```

Answer-generation user message — PersonaMem

```
You are a helpful assistant tasked with selecting the best answer to a user question,
     based solely on summarized conversation memories.

# CONTEXT:
The following are summarized facts and preferences extracted from prior user conversations
     . Use only these memories to answer the question.
```

```
{context}

# INSTRUCTIONS:
1. Carefully read and reason over the memory summary.
2. Evaluate each of the four answer choices (a) through (d).
3. Choose the single best-supported answer based on the information in memory.
4. Output ONLY the final choice in the format (a), (b), (c), or (d), placed directly after
      the token <final_answer>.

# IMPORTANT RULES:
- Your final answer **must appear after** the token <final_answer>.
- Your final answer **must use parentheses**, like (a) or (b).
- Do NOT list multiple choices. Choose only one.
- Do NOT include extra text after <final_answer>. Just output the answer.

# QUESTION:
{question}

# OPTIONS:
{options}

Final Answer:
<final_answer>
```

Answer-generation user message — BEAM

```
You are a knowledgeable and helpful AI assistant with access to memory context about a
     user.

# MEMORY TIMELINE (sorted by session date, earliest first):
{timeline}

# CONTEXT:
The following are relevant memories retrieved from prior conversations with the user.

{context}

# INSTRUCTIONS:
1. Carefully analyze the provided memory context and the memory timeline above.
2. Answer the question as thoroughly and accurately as possible based on the memories.
3. If the memories contain specific details (dates, names, numbers, preferences), include
     them in your answer.
4. For questions involving event order or timing, use the memory timeline to present
     events in chronological order.
5. If the memories are insufficient to fully answer the question, provide the best answer
     you can with available information and note any gaps.
6. Be detailed and comprehensive in your response.
7. The CURRENT DATE below is when the user asks this question --- treat it as "now". For "
     current/most recent/latest" questions, use the memory with the latest session_date at
      or before the current date; for duration or deadline questions, anchor relative time
      references to the current date.
8. The memories may contain CONTRADICTORY statements about this. Explicitly state that
     there is conflicting information, quote BOTH conflicting statements with their
     specific details, and then ask the user which one is correct --- do not pick a side
     or guess.
9. Only answer if the memories DIRECTLY contain the specific thing asked. Do NOT infer,
     guess, or substitute tangential/profile information. If the specific information is
     not present, reply exactly: "Based on the provided chat, there is no information
     related to [topic]."
# CURRENT DATE:
{question_date}

# QUESTION:
{question}

# ANSWER:
```

Placeholder notation. The LoCoMo and LongMemEval templates use {{NAME}} slots substituted by a literal single-pass renderer; the PersonaMem and BEAM templates use Python `str.format` slots ({context}, {question}, {options}, {timeline}, {question_date}), so the doubled braces in the BEAM judge templates below render as single braces to the model. {{MEMORIES}} receives the per-benchmark serialization described above; {{QuestionDate}} is the official LongMemEval question date, never the wall-clock date.

Table 25: Per-benchmark evaluation protocols. "Calls per answer" counts LLM judge calls; PersonaMem uses the official option-set match with no LLM judge.

Benchmark	Judge prompts	Calls per answer	Output parsing
LoCoMo	1 shared semantic judge	1	strict CORRECT/WRONG label
LongMemEval	5 official variants	1	official `yes`-substring rule
BEAM (nugget categories)	1 rubric template	one per rubric item	JSON score binned to 0/0.5/1
BEAM (event_ordering)	1 ordering template	1	JSON positions, Kendall τ_b
PersonaMem	none	0	option-set exact match

D.3.2 LLM-JUDGE EVALUATION

The judge sees only the question, the reference answer (or rubric), and the generated answer; the method identity and the retrieved context are never shown, and gold fields cannot reach the answer side by construction. Table 25 summarizes the per-benchmark protocols. LoCoMo category 5 is excluded before scoring. Only the BEAM judge calls carry a system message; the LoCoMo and LongMemEval judges receive a single user message. Judge decoding uses temperature 0.

LLM-judge system message — BEAM only

```
You are a precise and fair evaluation judge.
```

LLM-judge user message — LoCoMo

```
Your task is to label a generated answer to a memory question as exactly CORRECT or WRONG.

You will be given:
(1) a question,
(2) a gold answer,
(3) a generated answer.

The question asks about something one conversation participant should know about another
     participant from prior conversations.

# Grading rules

Be generous and semantic. Mark CORRECT when any of the following is true:
- The generated answer exactly matches the gold answer after ignoring case, punctuation,
     or harmless formatting differences.
- The generated answer is semantically equivalent to the gold answer, even if wording
     differs.
- The generated answer is more specific than the gold answer but entails the gold answer (
     for example, a pet name instead of the generic animal type, or a precise date instead
      of a relative time period).
- The generated answer contains all key facts from the gold answer and adds extra
     information that does not contradict the gold answer.
- For yes/no questions, the generated answer has the same polarity as the gold answer,
     even if it includes supporting detail.
- For time questions, different formats or equivalent relative/absolute dates are CORRECT
     when they refer to the same date or time period.
- For category/count questions, specific entity names are CORRECT if they imply the same
     category or count requested by the gold answer.

Mark WRONG only when the generated answer contradicts the gold answer, misses an essential
      gold fact, gives a different entity/date/number, or says no information is available
      when the gold answer contains a specific fact.

# Input

Question: {{QUESTION}}
Gold answer: {{GOLD_ANSWER}}
Generated answer: {{GENERATED_ANSWER}}

# Output

Return ONLY one string, with no explanation, no punctuation, no markdown:
CORRECT
or
```

```
WRONG
```

The LongMemEval judge prompt is selected by question category, with abstention questions (identifier suffix `_abs`) always using the abstention variant; all five official variants follow.

LLM-judge user message — LongMemEval (standard)

```
I will give you a question, a correct answer, and a response from a model. Please answer
    yes if the response contains the correct answer. Otherwise, answer no. If the
    response is equivalent to the correct answer or contains all the intermediate steps
    to get the correct answer, you should also answer yes. If the response only contains
    a subset of the information required by the answer, answer no.

Question: {{QUESTION}}

Correct Answer: {{GOLD_ANSWER}}

Model Response: {{GENERATED_ANSWER}}

Is the model response correct? Answer yes or no only.
```

LLM-judge user message — LongMemEval (temporal-reasoning)

```
I will give you a question, a correct answer, and a response from a model. Please answer
    yes if the response contains the correct answer. Otherwise, answer no. If the
    response is equivalent to the correct answer or contains all the intermediate steps
    to get the correct answer, you should also answer yes. If the response only contains
    a subset of the information required by the answer, answer no. In addition, do not
    penalize off-by-one errors for the number of days. If the question asks for the
    number of days/weeks/months, etc., and the model makes off-by-one errors (e.g.,
    predicting 19 days when the answer is 18), the model's response is still correct.

Question: {{QUESTION}}

Correct Answer: {{GOLD_ANSWER}}

Model Response: {{GENERATED_ANSWER}}

Is the model response correct? Answer yes or no only.
```

LLM-judge user message — LongMemEval (knowledge-update)

```
I will give you a question, a correct answer, and a response from a model. Please answer
    yes if the response contains the correct answer. Otherwise, answer no. If the
    response contains some previous information along with an updated answer, the
    response should be considered as correct as long as the updated answer is the
    required answer.

Question: {{QUESTION}}

Correct Answer: {{GOLD_ANSWER}}

Model Response: {{GENERATED_ANSWER}}

Is the model response correct? Answer yes or no only.
```

LLM-judge user message — LongMemEval (single-session-preference)

```
I will give you a question, a rubric for desired personalized response, and a response
    from a model. Please answer yes if the response satisfies the desired response.
    Otherwise, answer no. The model does not need to reflect all the points in the rubric
    . The response is correct as long as it recalls and utilizes the user's personal
    information correctly.

Question: {{QUESTION}}

Rubric: {{GOLD_ANSWER}}
```

```
Model Response: {{GENERATED_ANSWER}}

Is the model response correct? Answer yes or no only.
```

LLM-judge user message — LongMemEval (abstention)

```
I will give you an unanswerable question, an explanation, and a response from a model.
     Please answer yes if the model correctly identifies the question as unanswerable. The
      model could say that the information is incomplete, or some other information is
     given but the asked information is not.

Question: {{QUESTION}}

Explanation: {{GOLD_ANSWER}}

Model Response: {{GENERATED_ANSWER}}

Does the model correctly identify the question as unanswerable? Answer yes or no only.
```

LLM-judge user message — BEAM (rubric item)

```
You are an expert evaluator tasked with judging whether the LLM's response demonstrates
     compliance with the specified RUBRIC CRITERION.

## QUESTION:
{question}

## RUBRIC CRITERION (what to check):
{rubric_item}

## RESPONSE TO EVALUATE:
{response}

## SCORING:
- **1.0** = Fully satisfied: The response clearly and completely addresses this rubric
     criterion.
- **0.5** = Partially satisfied: The response addresses this criterion but is incomplete,
     vague, or only partially correct.
- **0.0** = Not satisfied: The response does not address this criterion at all, or is
     incorrect.

## SEMANTIC TOLERANCE:
Judge by meaning, not exact wording. Accept paraphrases and synonyms that preserve intent.
      Numbers/currencies/dates may appear in equivalent forms (e.g. "$68,000$", "68k").
     Treat them as equal when numerically equivalent.

Evaluate the response against ONLY this specific rubric criterion.

Return your evaluation as JSON:
{{"score": <0.0 or 0.5 or 1.0>, "reason": "<brief explanation>"}}
```

LLM-judge user message — BEAM

```
You are an expert evaluator. The question asked the user to list events in chronological
     order. The reference ordering of events is given below as a numbered list. Your task
     is to determine the order in which these same events appear in the response.

## QUESTION:
{question}

## REFERENCE ORDERING (correct chronological order):
{reference_ordering}

## RESPONSE TO EVALUATE:
{response}

## INSTRUCTIONS:
For each event in the reference list, determine its position (1-based) in the response. If
      an event is NOT mentioned in the response at all, assign position -1.

Return a JSON object mapping each event label to its detected position:
```

```
{{"positions": [{{"event": "<event label>", "position": <int or -1>}}, ...]}}
```

PersonaMem scoring. PersonaMem uses the official option-set protocol with no LLM judge: a response is correct exactly when the option letters after `<final_answer>` — or, failing that, the option letters in the whole response — form the singleton set containing the gold letter; ambiguous or missing labels score zero. The event-ordering score is coverage (fraction of reference events found) multiplied by the normalized Kendall τ_b over the found events, or half coverage when fewer than two events are found.