HiMem: A Hierarchical Multi-Granular System for Long-term Memory

Anonymous authors
Paper under double-blind review

Abstract

Long-term memory is critical for interactive agents to retain and utilize information accumulated over extended conversations. However, structurally organizing fragmented memories and accurately retrieving them remain significant challenges, particularly when relevant evidence lacks direct semantic similarity to the query. Existing systems still struggle to build a structural framework for evolving interaction records and to leverage structural information to bridge the semantic gaps. To address this challenge, we propose **HiMem**, a **Hi**erarchical Multi-granular **Mem**ory System with structure-aware multi-route retrieval, which organizes evolving memories and source records across multiple granularities and exploits their structural relations to recover complementary evidence. Specifically, HiMem organizes memories around verified anchors and constructs a multi-granular graph that preserves provenance of source records, building a foundation for multi-route retrieval. During retrieval, HiMem collects complementary candidates through multi-route retrieval and re-ranks them under a unified scoring mechanism that integrates semantic and structural signals. Experiments on LoCoMo show that HiMem achieves 92.60% average accuracy, outperforming the leading memory baseline, EverMemOS, by 2.15 percentage points. Furthermore, HiMem matches the performance of the full-context prompt method, while consuming merely 20.56% of the token cost.

1 Introduction

Large language models (LLMs) are increasingly deployed as interactive agents that engage with users over extended periods. To maintain continuity and consistency across such interactions, these agents require long-term memory to preserve and utilize information accumulated from previous conversations (Park et al., 2023; Sumers et al., 2024; Packer et al., 2024; Zhong et al., 2023; Chhikara et al., 2025). Simply extending the context window does not fully solve this problem: models often underuse evidence in long contexts, and performance degrades as relevant information becomes dispersed (Liu et al., 2024; Bai et al., 2024; An et al., 2024; Zhang et al., 2024; Hsieh et al., 2024). So it's important to construct structural memory systems to accurately retrieve necessary evidence. Early memory systems primarily relied on flattening the embedding space to store memories and dense or lexical retrieval to search historical records that directly match the current query in semantics (Robertson & Zaragoza, 2009; Karpukhin et al., 2020; Lewis et al., 2020; Chhikara et al., 2025; Liu et al., 2026). However, many long-term conversational questions require evidence distributed across different sessions. Flat embedding space and dense semantic matching fail to collect evidence across sessions, especially those lacking direct semantic similarity with the query. And it's difficult to recover the relations between these distant memories (Maharana et al., 2024; Wu et al., 2025). To connect such distributed evidence, recent memory systems have introduced structured representations, including entity graphs, semantic clusters, hierarchical memories, and event graphs (Rasmussen et al., 2025; Xu et al., 2025; Hu et al., 2026a; Tang et al., 2026; Hu et al., 2026b; Shu et al., 2026). These approaches demonstrate that relational organization is critical for memory systems to bridge the semantic gaps beyond isolated records.

Despite this progress, structurally organizing and accurately retrieving fragmented memories remain significant challenges. This challenge manifests in two critical aspects: **First, in memory construction**, existing systems rely heavily on semantic similarity to cluster evolving interaction records, overlooking underlying logical relations and real-world occurrence consistency. High se-

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.

mantic similarity does not guarantee that two memories describe the same occurrence, and conversely, memories of the same occurrence may have low surface similarity (Rasmussen et al., 2025; Hu et al., 2026b). Furthermore, it's important to retrieve appropriate historical context for memory evolution to avoid misjudgments, missed updates, or redundant changes during consolidation (Chhikara et al., 2025; Xu et al., 2025; Zhang et al., 2026). **Second, in memory retrieval**, a single dimension of memory representation is insufficient for global comprehension. Some memories require broader context to resolve ambiguous references. Existing systems lack effective mechanisms to retrieve complementary evidence from heterogeneous sources and rank them under a consistent criterion. Consequently, true evidence is often overshadowed by noise, failing to leverage the logical associations provided by the structured memory.

To address these challenges, we propose **HiMem**, a **Hi**erarchical Multi-Route **Mem**ory System that couples reliable structural memory organization with structure-aware evidence retrieval. Specifically, for memory construction, HiMem uses event nodes as unified anchors to organize atomic memories and performs a global retrieve-and-verify assignment based on occurrence compatibility. This mechanism links memories that refer to the same real-world occurrence while preventing semantically similar but distinct events from being conflated. The resulting event-centered structure further supports provenance-preserving memory consolidation and typed relation construction. For memory retrieval, HiMem performs structure-aware multi-route evidence retrieval over complementary representations and retrieval paths. It gathers complementary candidates through dense, event-mediated, relational, and lexical routes, thereby recovering evidence that may be missed by direct semantic matching alone. HiMem then re-ranks the retrieved evidence under a unified query-relevance criterion, where structural signals provide bounded support rather than overriding semantic relevance. In this way, the memory structure is not merely used for organization, but directly contributes to bridging semantic gaps during retrieval.

Experiments on the LoCoMo benchmark (Maharana et al., 2024) demonstrate the effectiveness of our approach. HiMem achieves 92.60% average accuracy, outperforming the leading memory baseline, EverMemOS, by 2.15 percentage points (Hu et al., 2026a). Moreover, HiMem matches the performance of the full-context reference while consuming merely 20.56% of its answer-stage input tokens, demonstrating that precise evidence selection is more effective than brute-force context inflation. Evidence-support analysis further reveals that event-mediated retrieval recovers an additional 18.45% of annotated evidence beyond direct dense retrieval, validating the efficacy of the structure-aware multi-route retrieval.

We summarize our main contributions as follows: (1) We introduce an event-anchored hierarchical memory architecture that organizes atomic memories according to occurrence compatibility rather than semantic similarity alone, providing a reliable structural basis for memory evolution, consolidation, and relational organization. (2) We develop a structure-aware multi-route retrieval mechanism that gathers complementary evidence through heterogeneous retrieval paths and reconciles them under a unified relevance-based ranking criterion, enabling structurally related evidence to be recovered without sacrificing direct query relevance. (3) Extensive experiments on long-term memory benchmarks show that HiMem improves both answering accuracy and token efficiency over representative memory systems, while fine-grained evidence-support analysis further demonstrates how event-mediated and structural retrieval recover evidence missed by direct retrieval.

2 Related Work

Structured Memory Organization and Evolution. External memory enables LLM agents to preserve information beyond the context window and use it in future interactions. MemGPT manages information between the active context and external storage (Packer et al., 2024), while Mem0 extracts factual memories and updates them through explicit memory operations(Chhikara et al., 2025). Earlier agent architectures likewise combine stored observations with reflection and planning, or maintain evolving user profiles over sustained interaction (Park et al., 2023; Zhong et al., 2023; Yao et al., 2023; Shinn et al., 2023). MemOS further provides a system-level framework for managing different forms of memory (Li et al., 2025). Recent studies organize growing interaction histories with more structured representations. Zep constructs episodic, entity, and community subgraphs (Rasmussen et al., 2025), and A-Mem dynamically links atomic memory notes (Xu et al., 2025). EverMemOS organizes memories into episodic and thematic structures, Mnemis constructs

hierarchical memory graphs, TraceMem forms narrative threads, and CompassMem represents interactions with event graphs (Hu et al., 2026a; Tang et al., 2026; Shu et al., 2026; Hu et al., 2026b). These studies demonstrate that structured memory can connect information distributed across interactions. However, the constructed structures often serve different stages of the memory lifecycle, such as storage, consolidation, or retrieval. Building upon this, HiMem introduces event nodes as unifying structural anchors that span the entire memory lifecycle, seamlessly integrating memory organization, evolution, and retrieval.

Structure-Aware Memory Retrieval. Memory retrieval determines which historical information is provided to an LLM for answering the current query. Early memory systems mainly rely on dense or lexical search to retrieve directly matched records (Lewis et al., 2020; Chhikara et al., 2025; Liu et al., 2026). Structured memory systems extend this process with graph traversal, hierarchical routing, and multi-scale retrieval. Zep combines semantic, lexical, and graph search with reranking (Rasmussen et al., 2025). EverMemOS retrieves evidence across different memory scales, Mnemis routes queries through a memory hierarchy, and CompassMem explores an event graph through iterative planning (Hu et al., 2026a; Tang et al., 2026; Hu et al., 2026b). These methods show that different retrieval routes can provide complementary evidence for long-term questions. HiMem follows this direction by retrieving evidence through dense, event-mediated, relational, and lexical routes. It further performs parallel ranking at two evidence levels: atomic memories are ranked using query and structural signals, while source segments are ranked using lexical and provenance support.

3 METHOD

In this section, we present a detailed description of the Hierarchical Multi-Route (HiMem) framework. First, we introduce the memory construction mechanism, which utilizes event nodes as unifying anchors to organize atomic memories based on occurrence compatibility, providing a reliable structural foundation for memory evolution. Then we propose the structure-aware multi-source retrieval method, which gathers complementary candidates through heterogeneous routes and ranks them under a unified query-relevance criterion. Detailed extraction prompts, graph schemas, and retrieval settings are provided in the appendix.

Let $\mathcal{D} = \{x_1, \ldots, x_T\}$ denote an interaction history. HiMem segments $\mathcal{D}$ into source records $\mathcal{C}$ and extracts atomic memories $\mathcal{M}$. It then constructs a memory graph $\mathcal{G} = (\mathcal{V}, \mathcal{R})$, where $\mathcal{V} = \mathcal{V}_M \cup \mathcal{V}_E$ contains memory and event nodes. Each memory retains a pointer to its source record. Given a query q, HiMem retrieves an evidence set $\mathcal{X}_q$ and generates the answer as

$$\mathcal{X}_q = \mathrm{Retrieve}(q; \mathcal{C}, \mathcal{M}, \mathcal{G}), \qquad y = \mathrm{LLM}(q, \mathcal{X}_q). \tag{1}$$

3.1 HIERARCHICAL MEMORY ORGANIZATION AND EVOLUTION

Hierarchical Memory Representation. HiMem constructs a multi-granular memory graph to capture information at different levels of abstraction. At the fine-grained level, self-contained atomic memories m_j serve as fundamental evidence units, each comprising text d_j, timestamp τ_j, embedding $\mathbf{h}_j$, status a_j, and a provenance pointer linking to the source records where m_j are extracted from. At the coarse-grained level, segmented source records c_i preserve the original dialogue context, with metadata of speakers, timestamps, and a short topic summary. To organize the segmented memories, event nodes $e \in \mathcal{V}_E$ are introduced as structural anchors that group atomic memories satisfying both semantic similarity and occurrence constraints. This hierarchical design structurally links distributed evidence across sessions, making a foundation for structural and multi-source retrieval. More details about source records segmentation and memory extraction are provided in Appendix C.

Occurrence-verified Organization. Although semantic similarity provides basic relevance, it is insufficient to determine actual occurrence consistency and capture complex structural relationships. To ensure the reliability of the hierarchical structure, HiMem performs a global retrieve-and-verify assignment. For each newly extracted memory m, HiMem first retrieves a set of candidate event nodes whose embedding similarity exceeds a threshold δ:

$$\mathcal{A}_{K_E}(m) = \mathrm{TopK}_{e \in \mathcal{V}_E} \left\{ \cos(\mathbf{h}_m, \mathbf{h}_e) \mid \cos(\mathbf{h}_m, \mathbf{h}_e) \geq \delta \right\}. \tag{2}$$

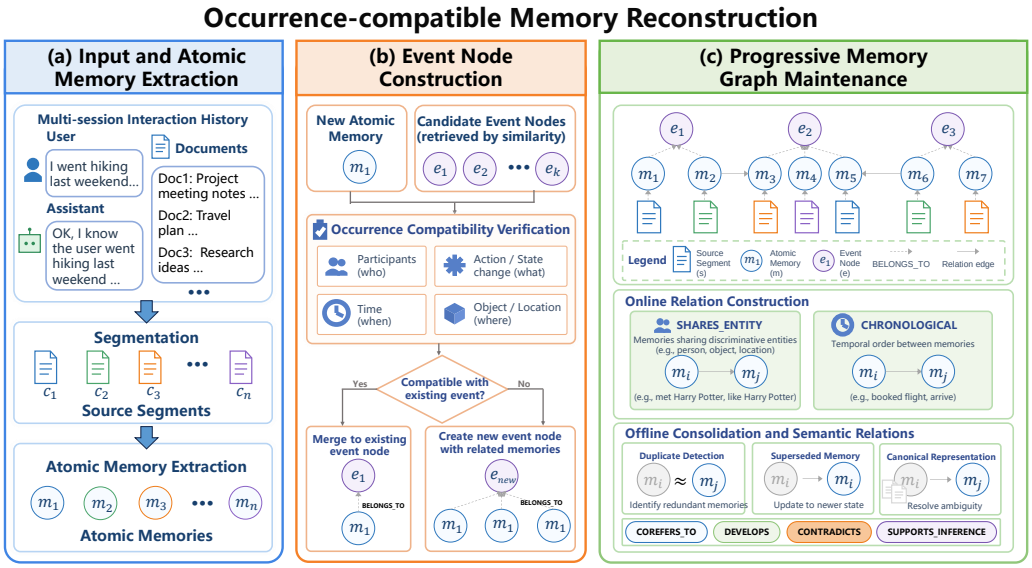

Figure 1: Hierarchical memory organization and evolution in HiMem. HiMem first segments the interaction history into source records and extracts atomic memories. Then, atomic memories are assigned to occurrence-verified event anchors, forming a multi-granular graph that supports provenance-preserving consolidation and relation construction.

An LLM-based compatibility verifier $v(m, e) \in \{0, 1\}$ then evaluates the multi-dimensional alignment between m and e, including participants, actions/states, objects/locations and temporal consistency, and recurrence. HiMem then assigns m to the most similar compatible candidate:

$$e^*(m) = \arg \max_{e \in \mathcal{A}_{K_E}(m)} \cos(\mathbf{h}_m, \mathbf{h}_e) \quad \text{s.t.} \quad v(m, e) = 1. \tag{3}$$

Unmatched memories form new event nodes based on their occurrences. This occurrence-verified mechanism ensures that the hierarchical graph is built upon a reliable relational foundation, preventing the entanglement of distinct real-world occurrences.

Provenance-preserving Memory Evolution. HiMem maintains the memory graph through two complementary stages: *online relation construction* and *deferred consolidation*. During online interaction, the system incrementally adds event membership edges and directly observable intra-event relations, such as shared entities and temporal ordering. In the deferred consolidation stage, HiMem performs event-local maintenance by labeling memories as active, duplicate, or superseded, and resolving ambiguous references into a canonical form. Crucially, this process is strictly provenance-preserving: the original memory and its provenance pointer are retained, enabling the on-demand restoration of the original dialogue context for fine-grained evidence verification. Furthermore, it identifies other typed relations—such as coreference, development, inference support, and contradiction. To avoid exhaustive pairwise comparisons, candidate relation generation is structurally constrained, and an LLM verifier retains only those relations explicitly supported by the underlying memories. The complete relation schema, consolidation rules, and candidate construction details are provided in Appendix D.

3.2 Structure-Aware Multi-Source Retrieval

Multi-Route Evidence Collection. To bridge semantic gaps and capture distributed evidence, HiMem retrieves candidates through complementary multi-route mechanisms at two distinct granularities. At the *memory level*, the candidate set $\mathcal{U}_M(q)$ is formed by the union of three paths: (1) *Dense retrieval* $\mathcal{U}_{\text{dense}}(q)$ directly matches the query against atomic memories based on semantic similarity; (2) *Event-mediated retrieval* $\mathcal{U}_{\text{event}}(q)$ matches the query against event anchors to capture the occurrence from a holistic perspective, subsequently collecting their member memories via graph edges; and (3) *Relational expansion* $\mathcal{U}_{\text{rel}}(q)$ expands outward from dense-hit seeds along the

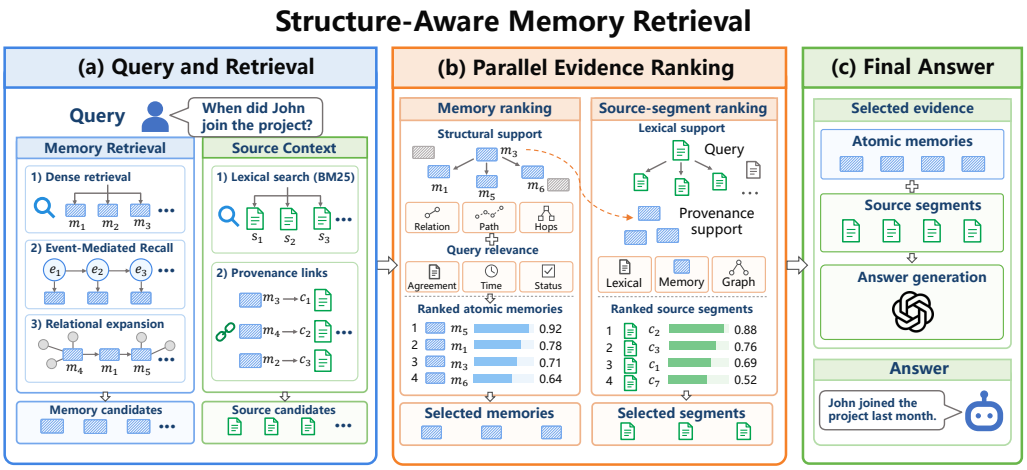

Figure 2: Structure-aware multi-source retrieval in HiMem. HiMem retrieves atomic memories via dense, event-mediated, and relational routes, and gathers source records through BM25 and provenance links. Both candidate sets are then independently re-ranked under their respective unified scoring mechanisms.

graph to retrieve topically related memories that may lack direct semantic overlap with the query. At the *source-record level*, HiMem aggregates candidates through lexical matching and provenance tracing: BM25 retrieves records via lexical overlap, while the provenance pointers directly load the corresponding source records. This multi-route collection ensures comprehensive evidence coverage across different granularities.

Unified Structure-Aware Evidence Ranking. Scores produced by heterogeneous retrieval routes are inherently incomparable. To reconcile these multi-source signals, HiMem establishes a unified ranking mechanism anchored by direct semantic relevance. Beyond the base semantic similarity$s_{\text{sem}}(m, q)$, we incorporate structural support $r(m, q)$ derived from event-query similarity or relational path confidence. The final memory score is formulated as:

$$s_M(m \mid q) = s_{\text{sem}}(m, q) + \lambda_g r(m, q) + \lambda_a I_{\text{agree}}(m, q) + b_{\text{date}}(m, q) - \lambda_s I_{\text{superseded}}(m). \quad (4)$$

where $I_{\text{agree}}(m, q)$ is a multi-source coverage bonus, $b_{\text{date}}(m, q)$ is a temporal alignment bonus, and $I_{\text{superseded}}(m)$ is a status penalty. By bounding the structural contribution through λ_g, HiMem ensures that structural signals calibrate rather than override semantic relevance. This mechanism effectively bridges the semantic gap, ensuring that structurally relevant evidence is not overshadowed by purely dense matches.

At the source-record level, HiMem independently ranks the candidate set $\mathcal{U}_C(q)$ by integrating four complementary signals: normalized memory support F_c, reciprocal-log BM25 rank B_c, query-term coverage L_c, and structural support G_c. The final source record score is formulated as:

$$s_C(c \mid q) = \alpha F_c + \beta B_c + \eta L_c + \zeta G_c. \quad (5)$$

By jointly optimizing lexical matching and memory provenance, this parallel ranking recovers granular contextual details omitted during atomic extraction. More details about the route scores and hyperparameters are provided in Appendix E.

4 Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of HiMem. The experimental evaluation consists of four main parts: **Experiment 1.** We compare HiMem with representative memory systems in answering accuracy. **Experiment 2.** We evaluate the trade-off between answering accuracy and answer-stage input token consumption. **Experiment 3.** We conduct ablation experiments to analyze the contribution of each component. **Experiment 4.** We examine whether the memory structure recovers useful evidence that direct retrieval misses.

Table 1: Accuracy (%) on LoCoMo using GPT-4.1-mini. Avg. Tokens denotes the average input tokens consumed by the answer stage. The best memory-system result is in bold.

Method	Avg. Tokens	LoCoMo				
		Multi Hop ↑	Temporal ↑	Open Domain ↑	Single Hop ↑	Overall ↑
Full context	22119	90.78	90.97	76.04	95.01	92.21
GPT-4.1-mini backbone						
ZEP	3384	80.50	70.40	67.71	90.96	82.99
Mem0	1524	66.67	23.05	51.04	67.78	57.21
LightMem	905	71.28	78.19	62.50	76.81	75.19
EverMemOS	3198	87.59	88.16	69.79	94.65	90.45
HiMem	4548	**89.72**	**91.59**	**75.00**	**95.96**	**92.60**
Full context	23958	92.91	94.70	86.46	97.38	95.32
Gemini-3.8-flash backbone						
ZEP	3384	86.52	72.90	70.83	87.63	83.31
Mem0	2583	79.43	61.06	64.58	77.29	73.51
LightMem	916	68.79	82.87	54.17	82.52	78.31
EverMemOS	3132	**90.78**	93.04	**78.82**	93.78	92.14
HiMem	5179	88.30	**93.15**	75.00	**96.91**	**93.18**

Datasets. We evaluate on two widely used public benchmarks for long-term conversational memory: LoCoMo (Maharana et al., 2024) and LongMemEval-S (Wu et al., 2025). LoCoMo consists of ten long-running, multi-session dialogues. Following standard evaluation protocols, we use its 1,540 non-adversarial questions, which cover single-hop, multi-hop, temporal, and open-domain reasoning. LongMemEval-S pairs each of 500 histories with a curated question designed to test single-session recall, cross-session synthesis, preference, knowledge update, or temporal reasoning. We report the LongMemEval-S results in Appendix F.1, and detailed dataset statistics and category definitions in Appendix A.

Experimental setup. Throughout all experiments, we adopt GPT-4.1-mini and Gemini-3.8-flash as the backbone models for memory construction and answer generation. Memory extraction follows each method's original decoding settings, while answer generation is performed with a temperature of 0 to ensure deterministic outputs. To ensure a fair comparison, we apply an identical LLM judge (GPT-4.1-mini) to evaluate all methods and retain each baseline's native memory construction and retrieval procedures. We primarily report the number of input tokens consumed during the answer generation stage as the efficiency metric. Comprehensive prompts, hyperparameters, and implementation details are provided in Appendix B.

Evaluation metrics and baselines. Throughout all experiments, we report accuracy by question category and overall, along with the average number of input tokens consumed during the answer generation stage. We also plot accuracy against answer-input tokens across retrieval budgets. Following established protocols, we employ an LLM-as-a-judge framework using GPT-4.1-mini. To ensure evaluation robustness, each generated answer is independently assessed three times, with correctness determined by a majority vote (at least two positive judgments). We compare HiMem with four representative memory systems, each reflecting distinct architectural paradigms: (1) **Mem0** (Chhikara et al., 2025), which relies on flat fact extraction and sequential updates without structural organization; (2) **Zep** (Rasmussen et al., 2025), which organizes temporal facts into an entity graph but lacks multi-route evidence gathering; (3) **LightMem** (Fang et al., 2026), which focuses on lightweight compression and offline updates but operates primarily on single-path semantic matching; and (4) **EverMemOS** (Hu et al., 2026a), the current state-of-the-art, which organizes memories into episodic and thematic structures but lacks a unified scoring mechanism to reconcile heterogeneous retrieval signals. Additionally, we include a **Full-Context** reader as a high-cost reference to highlight the efficiency gains of our structured retrieval approach. **Experiment 1. Answering Accuracy** In this subsection, we compare HiMem with baselines in terms of answering accuracy on LoCoMo. As shown in Table 1, HiMem achieves the highest overall accuracy across both backbones, demonstrating superior effectiveness and stability. For the GPT-4.1-mini backbone, HiMem

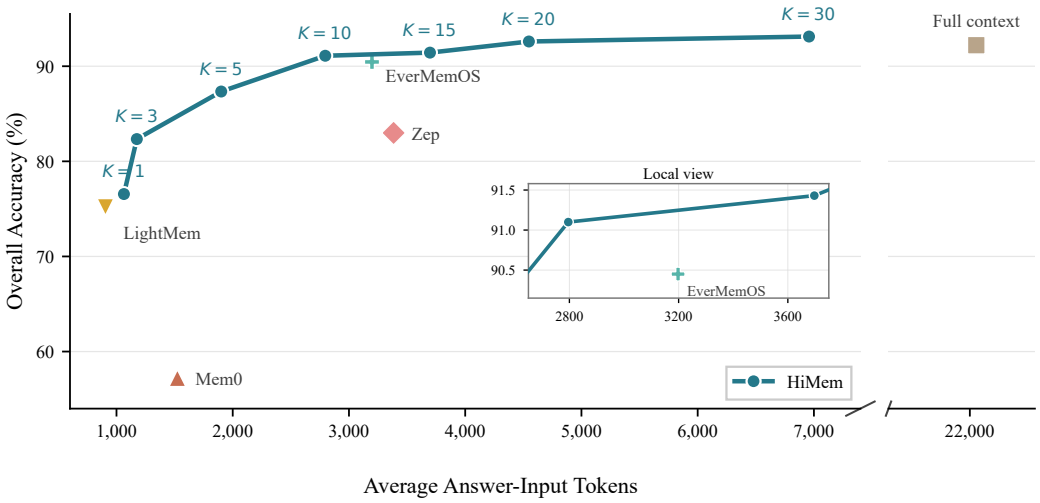

Figure 3: Accuracy versus average input token consumption in the answer stage on LoCoMo. Labels on the HiMem curve denote the memory retrieval budget K; the source-chunk budget C is increased jointly. Baselines are evaluated under their standard settings. Notably, HiMem consistently outperforms all baselines across all token budgets, dominating the preferred upper-left region.

achieves 92.60% overall accuracy, outperforming the strongest baseline EverMemOS by 2.15 percentage points. For the Gemini-3.8-flash backbone, HiMem achieves 93.18%, surpassing EverMemOS by 1.04 percentage points. These consistent gains across different model scales confirm the robustness of our approach. Category-wise analysis reveals that HiMem yields substantial improvements on multi-hop, temporal, and open-domain questions (e.g., +5.21 points on open-domain for GPT), highlighting the advantage of HiMem's hierarchical organization in capturing complex relational dependencies. Notably, HiMem slightly exceeds the full-context reference (92.60% vs. 92.21%) while consuming only 20.56% of its answer-input tokens. This result demonstrates that our structured evidence selection mechanism is more effective than placing the complete history in the answer context, avoiding potential noise while maintaining high-quality evidence retrieval.

Experiment 2. Accuracy–Token Trade-off In this subsection, we investigate whether the accuracy gains of HiMem rely on simply expanding the answer-stage context. We jointly vary the memory retrieval budget (K) and the source-segment budget (C), evaluating the configurations $(K, C) \in \{(1,1),(3,1),(5,2),(10,3),(15,4),(20,5),(30,8)\}$. Baselines are evaluated under their default settings. We adopt the answer-stage input token count as a unified cost metric, as memory systems employ different representations and processing pipelines, which make token consumption during construction and retrieval difficult to compare directly. In contrast, answer-stage tokens directly reflects the evidence delivered to the LLM and affects inference latency. As shown in Figure 3, HiMem achieves steady accuracy improvements as the retrieval budget increases, consistently outperforms all baselines across all token budgets. Specifically, at $K = 3$, HiMem achieves 82.34% accuracy using only 1,175 tokens, surpassing the lightweight baseline, LightMem by 7.15 percentage points, at comparable token cost. Furthermore, it matches Zep's accuracy (82.99%) while reducing the answer-stage token consumption by 65.28%. At $K = 10$, HiMem reaches 91.10% with 2,796 tokens, outperforming EverMemOS by 0.65 percentage points while using 12.57% fewer tokens. At $K = 20$, HiMem achieves 92.60% accuracy, surpassing the full-context reference while reducing token consumption by 79.44%. These results demonstrate that the performance gains of HiMem stem from the precise selection of highly useful evidence, rather than merely inflating the context length.

Experiment 3. Ablation Study In this subsection, we conduct ablation experiments to examine the contribution of each component in HiMem. We first evaluate four variants of memory construction and maintenance. Specifically, *w/o event verification* assigns memories by semantic similarity without occurrence verification; *w/o event structure* removes event nodes and the corresponding retrieval route; *w/o consolidation* skips memory-state and canonical-form updates; and *w/o typed relations* removes the online and offline relation edges. We then compare different combinations of dense,

Table 2: Ablation results on LoCoMo. Avg. Tokens denotes the average input tokens consumed by the answer stage. Δ denotes the change in overall accuracy relative to HiMem.

Variant	Avg. Tokens	Multi Hop	Temporal	Open Domain	Single Hop	Overall	Δ
HiMem	4,548	**89.72**	**91.59**	**75.00**	95.96	**92.60**	–
Memory construction and maintenance							
w/o event verification	4,518	89.72	89.41	68.75	95.96	91.62	-0.97
w/o event structure	4,407	85.46	89.41	71.88	95.72	90.84	-1.75
w/o consolidation	4,476	89.72	90.97	69.79	**96.55**	92.21	-0.39
w/o typed relations	4,536	89.72	90.65	68.75	96.08	91.75	-0.84
Retrieval components							
dense only	1,681	85.11	88.47	70.83	89.42	87.14	-5.45
BM25 only	2,740	72.70	79.75	64.58	92.27	83.96	-8.64
dense + BM25	4,445	86.88	91.28	70.83	95.96	91.30	-1.30
dense + graph	1,684	85.46	85.98	70.83	89.89	86.82	-5.78
BM25 + graph	4,593	89.36	90.65	70.83	96.31	91.95	-0.65
w/o structural scoring	3846	86.88	90.03	70.83	96.20	91.62	-0.97

lexical, and graph retrieval to analyze the contribution of each retrieval route. All variants use the same answer model and evaluation protocol.

Memory construction. Among the construction-side variants, removing the event structure causes the largest accuracy drop (-1.75 points), particularly on multi-hop (-4.26 points) and open-domain questions (-3.12 points). This confirms that the event-anchored hierarchical structure, built upon occurrence compatibility, is essential for aggregating distributed evidence across different time spans and dialogue sessions. Disabling event verification reduces accuracy by 0.97 points. This demonstrates that verifying occurrence compatibility provides a more reliable relational foundation than relying solely on semantic similarity clustering. Typed relations contribute 0.84 points to overall accuracy by encoding fine-grained semantic associations, while consolidation contributes 0.39 points by resolving ambiguity and state updates.

Retrieval components. The retrieval ablations reveal strong complementarity between memory-level evidence (dense), segment-level lexical evidence (BM25) and graph-based structural evidence. Removing either side leads to substantial degradation: dense-only and BM25-only drop by 5.45 and 8.64 points, respectively. This complementarity is further evidenced by the large gains when combining the two levels: adding BM25 to dense retrieval improves accuracy by 4.16 points, while adding dense retrieval to BM25 yields an improvement of 7.34 points. Moreover, adding graph-based expansion to BM25 yields an even larger gain of 7.99 points. Beyond complementarity, structural retrieval plays a critical role in re-ranking memory-level evidence. The higher accuracy of BM25 + graph shows that graph-based expansion effectively recovers evidence lacking direct surface similarity. More importantly, adding graph retrieval to the dense + BM25 combination yields a further 1.30-point improvement ($91.30 \rightarrow 92.60$), reaching the best overall performance. Removing the structure-aware scoring while retaining the same retrieval routes and common semantic relevance reduces overall accuracy by 0.97 points, with the largest degradation on open-domain (-4.17) and multi-hop (-2.84) questions. These results demonstrate that our structure-aware reranking mechanism successfully recover structural support signals, enabling the model to prioritize evidence with stronger logical and structural relevance.

Experiment 4. Evidence-Support Analysis In this subsection, we conduct a fine-grained analysis to examine how HiMem assembles answer-supporting evidence through its heterogeneous retrieval routes. The LoCoMo dataset provides explicit annotations of the ground-truth evidence turns required to answer each question. To evaluate retrieval effectiveness, we employ an LLM-based evidence-equivalence judge to determine whether a retrieved atomic memory semantically aligns with the target evidence. This process yields 2,358 gold-evidence instances across 1,535 eligible questions. We attribute each instance to distinct retrieval categories based on metadata: (1) *dense support*: instances containing at least one match from the dense retrieval route; (2) *event-only support*: instances matched exclusively through the event-mediated route, without any dense-sourced match; (3) *raw-context-only support*: instances where no atomic memory matches, but the annotated source turn is present within a retrieved source segment; and (4) *unsupported*: instances failing to

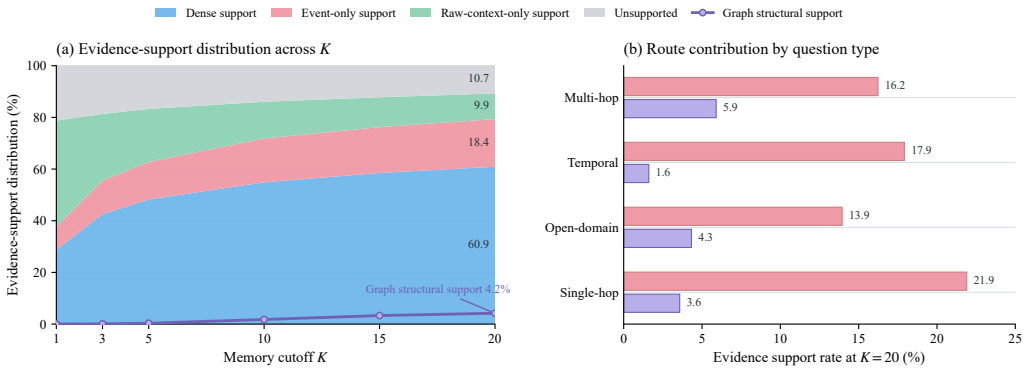

Figure 4: Evidence-support analysis of HiMem on LoCoMo. (a) Distribution of the 2,358 annotated gold-evidence instances as the memory cutoff K increases. Filled regions form an incremental, mutually exclusive decomposition; graph-expanded support is overlaid because it can coincide with dense- or event-supported evidence. Raw contexts are fixed to those returned by the $(K, C) = (20, 5)$ configuration. (b) Event-only and graph-expanded support across question types at $K = 20$.

meet any of the above criteria. Additionally, because graph expansion can introduce new candidates that coexist with other routes, we report *graph structural support* as an independent, overlapping signal that provides structural reinforcement during unified re-ranking. For this analysis, we utilize the $(K, C) = (20, 5)$ configuration, keeping the source records fixed while varying the memory truncation depth at $K \in \{1, 3, 5, 10, 15, 20\}$.

As illustrated in Figure 4(a), the proportion of evidence supported by retrieved atomic memories steadily increases from 37.79% at $K = 1$ to 79.35% at $K = 20$. At the maximum depth of $K = 20$, dense-sourced memories account for 60.90% of the annotated evidence. Notably, the event-mediated route contributes an exclusive 18.45% of evidence, without any equivalent dense-sourced match. This sizable non-dense contribution demonstrates that event nodes provide an additional retrieval path beyond direct semantic matching. Furthermore, the presence of graph structural support validates the effectiveness of our structure-aware scoring mechanism. Source records reconstruction further supplies 9.92% of the evidence absent from the retrieved atomic memories, bringing the combined support rate to 89.27%, which highlights the strong complementarity of multi-source evidence. Graph-expanded memories support 4.20% of all annotated evidence at $K = 20$. This demonstrates that structure-aware ranking successfully enables memories with low surface similarity to enter the final candidate pool based on reliable structural relations, effectively bridges semantic gaps. As shown in Figure 4(b), graph support is most frequent for multi-hop questions (5.90%), followed by open-domain questions (4.33%). Together with the ablation results in Section 3.2, these findings show two complementary roles of the event-anchored memory graph: event retrieval recovers evidence beyond direct semantic matches, while relational expansion provides calibrated structural support for more compositional questions.

5 Conclusion

Long-term memory is essential for interactive agents, yet evidence distributed across evolving sessions remains difficult to organize and retrieve effectively. To address this challenge, we proposed HiMem, a hierarchical memory system that organizes atomic memories around occurrence-verified event anchors and performs structure-aware multi-route evidence retrieval. By coupling reliable memory organization with complementary candidate retrieval routes, HiMem enables structurally related evidence to be recovered even when direct semantic matching is insufficient. Experiments on LoCoMo demonstrate that HiMem outperforms the strongest evaluated memory baseline by 2.15 percentage points, while event-mediated retrieval recovers an additional 18.45% of annotated evidence beyond dense retrieval. These results highlight the importance of jointly designing memory structure and retrieval, leveraging the structural information to recover complementary evidence.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods and interpret results. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, or support qualitative and thematic data analysis. Generating synthetic data sets and cleaning, assisting with translation, and reformatting datasets are not applicable to this work. Additionally, we used generative AI tools to suggest experimental parameters, create or edit software code, and edit the research paper to improve readability. We have reviewed all AI-assisted work. Specifically, we manually verified all LLM-assisted code and tested LLM-suggested experimental parameters. LLM-generated text was carefully edited to ensure scientific accuracy, rigor, and fidelity to our original research ideas. We take full responsibility for the final content of this work, including all text, scientific claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We evaluate HiMem on the public releases of LoCoMo and LongMemEval-S and do not collect new data from human participants. Long-term memory systems may retain sensitive personal information, retrieve outdated or incorrect claims, or expose details beyond what a user intended to share. However, this is a common problem for all memory-based systems. Deployment with real user conversations would require appropriate consent, access controls, retention and deletion policies, and safeguards for any external model services that process the conversations. Our results on two benchmarks should not be taken as evidence that these requirements are met in a deployed system.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility and transparency of our work, we have made comprehensive efforts to document and share all necessary resources. Upon acceptance, we will provide an anonymous link to our source code repository, which includes the complete implementation of the HiMem framework, data preprocessing scripts, configuration files, and evaluation pipelines.

Detailed descriptions of the datasets used (LoCoMo and LongMemEval-S), including our specific data processing, filtering, and splitting steps, are thoroughly provided in Appendix A. Furthermore, Appendix B offers a transparent account of the baseline implementations, including exact version control (commit hashes) and the shared experimental environment, ensuring fair and reproducible comparisons.

For our proposed method, Appendix C, Appendix D, and Appendix E comprehensively detail the algorithmic workflows, architectural designs, and all hyperparameter settings (which are systematically summarized in Table 5). Finally, our rigorous evaluation protocol, including the LLM-as-a-judge setup with multiple independent calls and majority voting to ensure robustness, is explicitly documented in Appendix F.7. Collectively, these materials provide all the necessary components to faithfully reproduce the results reported in this paper.

REFERENCES

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. L-eval: Instituting standardized evaluation for long context language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14388–14411, 2024.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. 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)*, pp. 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. RAGAs: Automated evaluation of retrieval augmented generation. In Nikolaos Aletras and Orphee De Clercq (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 150–158, St. Julians, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-demo.16. URL https://aclanthology.org/2024.eacl-demo.16/.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. Lightmem: Lightweight and efficient memory-augmented generation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 98706–98729, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a05b72653ec5b473732129829ae04195-Paper-Conference.pdf.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: what's the real context size of your long-context language models? *CoRR*, abs/2404.06654, 2024. doi: 10.48550/ARXIV.2404.06654. URL https://doi.org/10.48550/arXiv.2404.06654.

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)*, pp. 45836–45853, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2125. URL https://aclanthology.org/2026.acl-long.2125/.

Yuyang Hu, Jiongnan Liu, Jiejun Tan, Yutao Zhu, and Zhicheng Dou. Memory matters more: Event-centric memory as a logic map for agent searching and reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 22389–22407, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1123. URL https://aclanthology.org/2026.findings-acl.1123/.

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

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 H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Zhiyu Li, Chenyang Xi, Chunyu Li, Ding Chen, Boyu Chen, Shichao Song, Simin Niu, Hanyu Wang, Jiawei Yang, Chen Tang, Qingchen Yu, Jihao Zhao, Yezhaohui Wang, Peng Liu, Zehao Lin, Pengyuan Wang, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhen Tao, Huayi Lai, Hao Wu, Bo Tang, Zhengren Wang, Zhaoxin Fan, Ningyu Zhang, Linfeng Zhang, Junchi Yan, Mingchuan Yang, Tong Xu, Wei Xu, Huajun Chen, Haofen Wang, Hongkang Yang, Wentao

Zhang, Zhi-Qin John Xu, Siheng Chen, and Feiyu Xiong. Memos: A memory os for ai system, 2025. URL https://arxiv.org/abs/2507.03724.

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

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

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

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024. URL https://arxiv.org/abs/2310.08560.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025. URL https://arxiv.org/abs/2501.13956.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009. ISSN 1554-0669. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

Yiming Shu, Pei Liu, Tiange Zhang, Ruiyang Gao, Jun Ma, and Chen Sun. TraceMem: Weaving narrative memory schemata from user conversational traces, 2026. URL https://arxiv.org/abs/2602.09712.

Theodore Sumers, Shunyu Yao, Karthik R Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=1i6ZCvflQJ. Survey Certification, Featured Certification.

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)*, pp. 23914–23928, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1096. URL https://aclanthology.org/2026.acl-long.1096/.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 86809–86836, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/d813d324dbf0598bbdc9c8e79740ed01-Paper-Conference.pdf.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents, 2025. URL https://arxiv.org/abs/2502.12110.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Dylan Zhang, Yanshan Lin, Zhengkun Wu, Yihang Sun, Bingxuan Li, Dianqi Li, and Hao Peng. Useful memories become faulty when continuously updated by llms, 2026. URL https://arxiv.org/abs/2605.12978.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞Bench: Extending long context evaluation beyond 100K tokens. 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)*, pp. 15262–15277, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.814. URL https://aclanthology.org/2024.acl-long.814/.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory, 2023. URL https://arxiv.org/abs/2305.10250.

A Datasets and Evaluation Splits

We evaluate HiMem on the public evaluation releases of two complementary long-term conversational-memory benchmarks. LoCoMo tests question answering over extended two-speaker dialogues, with supporting evidence often distributed across sessions. LongMemEval-S instead associates each question with a long, timestamped user–assistant history and includes capabilities that are not explicitly isolated in LoCoMo, such as recalling assistant-provided information and tracking updates to user information. Neither release provides an in-domain training split: we use no benchmark examples for model training and evaluate all examples in the subsets specified below.

Table 3 reports statistics recomputed directly from the released `locomo10.json` and `longmemeval_s_cleaned.json` files. The counts describe the released JSON records before our retrieval pipeline or the message-splitting operation described below.

Table 3: Statistics of the evaluation sets used in this paper. "Released Q" and "Eval. Q" denote the numbers of questions in the released file and in our evaluation subset, respectively. A LongMemEval-S history is compiled for one question; its session and message counts are therefore averaged over the 500 question-specific histories.

Dataset	Histories	Released Q	Eval. Q	Avg. sessions	Avg. messages	Evaluated types
LoCoMo	10	1,986	1,540	27.2	588.2	Four non-adversarial types
LongMemEval-S	500	500	500	47.7	493.5	Six `question_type` labels

A.1 LoCoMo

LoCoMo is a benchmark of synthetically generated, human-annotated dialogues that unfold over multiple sessions (Maharana et al., 2024). We use the public LoCoMo-10 release, which contains

ten dialogues, 272 non-empty sessions, 5,882 messages, and 1,986 questions. Following the four-category protocol used by the memory-system baselines in our comparison (Chhikara et al., 2025; Hu et al., 2026a), we exclude the 446 adversarial questions whose answers are deliberately absent from the dialogue. The resulting evaluation set contains 1,540 questions: 282 multi-hop, 321 temporal, 96 open-domain, and 841 single-hop questions. These labels correspond, respectively, to category IDs 1, 2, 3, and 4 in the released JSON file; we state the mapping explicitly because the numeric IDs do not follow the order in which the categories are introduced in the original paper. Each non-adversarial question has a free-form reference answer, and all but four have one or more supporting dialogue-turn identifiers. We use the full 1,540-question subset for answer evaluation and do not derive a development set from it.

A.2 LongMemEval-S

LongMemEval evaluates whether an assistant can retain, integrate, and update information over long, timestamped user–assistant histories (Wu et al., 2025). We use the complete 500-question LongMemEval-S release. Each question is paired with a separately compiled history containing 47.7 sessions and 493.5 messages on average (23,867 session occurrences and 246,750 message occurrences across the release). The official paper characterizes LongMemEval-S as approximately 115K tokens per question; because that value depends on the tokenizer, we report session and message counts rather than mixing token and whitespace-delimited word statistics.

The released `question_type` field has six labels: single-session user recall (70 questions), single-session assistant recall (56), single-session preference (30), multi-session reasoning (133), knowledge update (78), and temporal reasoning (133). These labels refine the benchmark's five higher-level memory abilities; in addition, 30 questions distributed across four of the six labels are abstention variants, identified by the `_abs` suffix in their question IDs. We retain all 500 questions. During ingestion, message content longer than 8,000 characters is split into ordered parts without discarding text; each part retains the identifier of its original message so that evidence provenance remains recoverable.

B Baselines and Experimental Settings

Following the conventions of prior systems papers, we separate the method-specific hyperparameters (detailed in Appendix E) from the shared evaluation environment reported here. All methods process the same ordered conversations and questions. To ensure a fair comparison, we retain each baseline's native memory representations and retrieval algorithms; the answer model thus consumes the evidence naturally returned by each system, rather than an artificially matched number of heterogeneous records.

B.1 Baseline Implementations

To ensure reproducibility and mitigate the impact of rapid upstream updates, we freeze each baseline at a paper-aligned revision selected prior to evaluation, rather than using their latest development branches. Specifically, we use Mem0 v0.1.95 (commit `ad98f542`), Graphiti v0.5.1 for Zep (commit `77cb67c`), LightMem (commit `443d9c6`), and EverMemOS v1.1.0 (commit `77c416c`). These versions identify the upstream codebases; the controlled model substitutions and retrieval settings applied in our evaluation are detailed below and in Table 4.

Full context. The full-context approach serves as an upper-bound reference by incorporating the complete preceding conversation and questions directly into the prompt, evaluating the model's reasoning capabilities without any external memory mechanisms. In our evaluation, it is treated as a high-input-cost reference rather than a deployable memory-system baseline, and its input cost is measured using the same answer-stage accounting applied to all methods.

Mem0. Mem0 introduces a dynamic memory management system that extracts factual memories from dialogue turns and maintains them through explicit memory operations (i.e., ADD, UPDATE, DELETE, NONE). This mechanism allows the system to continuously refine its understanding of user interactions by updating or discarding outdated information in a flat vector store. For our

Table 4: Common model and evaluation settings.

Parameter	Value
Construction backbone	GPT-4.1-mini / Gemini-3.8-flash
Embedding model	`text-embedding-3-small`
Embedding dimension	1,536
Answer backbone	GPT-4.1-mini / Gemini-3.8-flash
Correctness judge	GPT-4.1-mini
Separate judge calls per answer	3
Majority threshold	At least 2 of 3 positive judgments

evaluation, we utilize its open-source implementation with graph storage disabled to focus on its core vector-based retrieval.

Zep (Graphiti). Zep constructs a temporal knowledge graph to manage long-term memory, organizing interactions into episodes and linking them via entity and fact edges. This architecture supports hybrid retrieval combining semantic, lexical, and graph traversal, enabling the system to capture the evolution of information and relationships over time. We evaluate its open-source Graphiti implementation, where episodes are inserted sequentially and namespaced by group identifier in Neo4j. At query time, it retrieves entity nodes using reciprocal-rank fusion and fact edges using cross-encoder reranking, with community updates disabled to align with standard conversational-memory evaluation protocols.

LightMem. Inspired by the Atkinson-Shiffrin human memory model, LightMem proposes a lightweight and efficient memory framework comprising three complementary stages: a pre-compressing sensory memory to filter redundant tokens, a topic-aware short-term memory for transient information processing, and a long-term memory with a sleep-time update mechanism that decouples expensive consolidation from online inference. In our implementation, we retain its LoCoMo-specific extraction prompt and configure LLMLingua-2 for pre-compression. During the offline consolidation phase, it retrieves update candidates based on semantic similarity thresholds to resolve inconsistencies and strengthen cross-knowledge connections.

EverMemOS. EverMemOS presents a hierarchical memory architecture inspired by operating systems, organizing interaction histories into episodic memories and thematic structures. It employs an agentic retrieval mechanism that routes queries across different memory scales, balancing the breadth of lexical search with the depth of structured relational retrieval (Hu et al., 2026a). We run its released local service in group-scoped, agentic-retrieval mode and request the top ten episodic memories. Ingestion and search remain single-perspective for each human–assistant history, and the returned episode text is passed to the common answer pipeline without rewriting its internal hierarchy.

B.2 Common Models and Evaluation Protocol

In this subsection, we describe the models, input handling, and evaluation protocol shared across methods. The main comparison uses GPT-4.1-mini for memory construction, answer generation, and judging. Each baseline retains its original construction prompt and decoding settings; HiMem and Mem0 use a construction temperature of 0.1. For all methods, answer generation uses a temperature of 0. All systems use `text-embedding-3-small` with 1,536-dimensional embeddings, except Zep, whose native interface uses 1,024 dimensions. Table 4 summarizes these settings. Method-specific graph and retrieval parameters are reported in Table 5.

We retain all message content in LoCoMo. For LongMemEval-S, messages exceeding 8,000 characters are split into ordered pieces without discarding content. When preparing HiMem's answer context, we select a query-centered excerpt of at most 8,000 characters from each retrieved source record. We measure answer-stage cost as the average number of input tokens supplied to the answer model. For answer evaluation, the judge receives the question, reference answer, and generated

answer, but not the method identity. We call the judge three times for each answer and determine correctness by majority vote. Individual judgments are also retained for paired analyses. Since LLM-based judgments may contain errors, we report this protocol explicitly rather than treating a single judgment as definitive (**?**An et al., 2024). Evidence-support analysis uses a separate judge to assess whether a retrieved atomic memory is semantically equivalent to an annotated source turn. These evidence judgments are used only to analyze retrieval and do not replace answer-correctness judgments (Es et al., 2024).

C Memory Formation and Details

C.1 Topic-Aware Segmentation

Long conversations should not be processed as a single block for memory extraction. Unrelated sessions lead to over-compression and the loss of fine-grained details, while individual messages lack sufficient context for pronoun and event resolution. Adopting a two-stage segmentation strategy inspired by LightMem (Fang et al., 2026), HiMem first detects semantic topic boundaries and subsequently enforces temporal continuity constraints.

Let a *turn* x_i denote the concatenation of a user message and its immediate assistant reply, and let $\mathbf{v}_i$ be its dense embedding. For coarse segmentation, the user message from each turn is encoded by an attention-based segmenter (e.g., LLMLingua-2). The cross-attention is averaged across selected transformer layers and all heads, and then aggregated from the tokens in turn i to the tokens in each preceding turn $j < i$. These mean attention scores are row-normalized across all preceding turns to form the attention matrix M:

$$M_{ij} = \frac{\mathrm{MeanAttn}(x_i, x_j)}{\sum_{k<i} \mathrm{MeanAttn}(x_i, x_k)}, \quad \text{for } j < i. \tag{6}$$

Let $A_i = M_{i,i-1}$ denote the attention weight assigned to the immediately preceding turn. Local maxima in this sequence yield coarse boundary candidates: $\mathcal{B}_{\mathrm{att}} = \{i \mid A_{i-1} < A_i > A_{i+1}\}$. A local maximum indicates a semantic shift, where the current turn shifts its focus to the immediately preceding turn rather than the broader earlier context, signaling the onset of a new topic. If the attention model is unavailable, M is alternatively constructed using non-negative pairwise cosine similarities of turn embeddings, followed by the identical row normalization.

Fine-grained boundaries are identified via adjacent-turn similarity. Rather than relying on a fixed, dataset-specific threshold, we dynamically scan a candidate set $\Theta = \{0.20, 0.25, \ldots, 0.50\}$ and select the first threshold that yields at least one boundary:

$$\theta^* = \min\{\theta \in \Theta \mid \mathcal{B}_{\mathrm{sem}}(\theta) \neq \emptyset\}, \tag{7}$$

$$\mathcal{B}_{\mathrm{sem}}(\theta) = \{i \mid \cos(\mathbf{v}_{i-1}, \mathbf{v}_i) < \theta\}. \tag{8}$$

The union of coarse and fine candidates is deduplicated by enforcing a minimum spacing of two turns, and subsequently mapped back to the original message indices.

Crucially, HiMem inserts an inviolable temporal boundary whenever the time gap between two consecutive messages exceeds a predefined threshold (e.g., 6 hours):

$$\mathcal{B}_{\mathrm{time}} = \{i \mid \tau_i - \tau_{i-1} > \tau_{\mathrm{gap}}\}. \tag{9}$$

This hard constraint prevents high topical similarity from erroneously merging temporally disjoint occurrences of the same topic. Finally, a post-merging step resolves over-segmentation: any fragment containing fewer than predefined minimum thresholds of user messages (e.g., 2) or total messages (e.g., 4) is merged into the temporally closest adjacent fragment. Each resulting source record c_i preserves the original message order, roles, and timestamps, serving as the coarse-grained unit for provenance-preserving memory evolution. The topic summary z_i is subsequently generated from c_i during the memory construction phase.

C.2 Atomic Memory Extraction

To construct the fine-grained level of the hierarchical memory graph, an LLM extracts self-contained atomic memories from each source record. The extraction model processes the original messages,

speaker identities, and the record's time range to generate statements with explicit subjects and resolved temporal expressions:

$$\mathcal{M}_i = \text{Extract}_\phi(c_i, [\tau_i^-, \tau_i^+]), \qquad \mathcal{M} = \bigcup_i \mathcal{M}_i. \tag{10}$$

The extraction schema employs eight descriptive tags—PREFERENCE, PERSONAL, HEALTH, PROFESSIONAL, PLAN, EPISODIC, OPINION, and EMOTION—followed by an absolute date and the memory text. These tags standardize the representation and are retained as metadata. Crucially, to avoid heuristic bias, they are not assigned manual retrieval weights; instead, query relevance is learned holistically from the semantic embeddings of the complete memory text.

Each atomic memory is formally represented as a tuple:

$$m_j = (u_j, d_j, \tau_j, \mathbf{h}_j, \ell_j, a_j, \bar{d}_j, \rho_j), \tag{11}$$

where u_j is a unique identifier, d_j is the extracted text, τ_j is its timestamp, $\mathbf{h}_j$ is its dense embedding, ℓ_j is the category tag, $a_j \in \{\text{active}, \text{duplicate}, \text{superseded}\}$ is its lifecycle status, $\bar{d}_j$ is an optional canonical form for ambiguous references, and ρ_j is the provenance pointer to the source records. Auxiliary metadata such as the topic identifier, topic summary, user identifier, and creation time are also stored. The original source record is preserved in a separate store; thus, extraction is not a lossy replacement of the conversation. The pointer ρ_j ensures that the exact local context can be restored on demand during retrieval and memory evolution.

D GRAPH CONSTRUCTION AND CONSOLIDATION DETAILS

D.1 EVENT VERIFICATION AND CREATION

Semantic embedding similarity alone is insufficient for proposing event anchors, as memories may share entities or topics yet describe distinct real-world occurrences. To ensure the reliability of the hierarchical structure, HiMem employs a rigorous occurrence-verified assignment mechanism. For a candidate memory m and an event anchor e, an LLM-based compatibility verifier evaluates their multi-dimensional alignment, formulated as:

$$v(m, e) = \mathbb{I}[P \wedge A \wedge O \wedge T \wedge \neg R], \tag{12}$$

where P, A, O, T denote compatibility of participants, action/state transition, object/location, and temporal consistency, respectively, and R indicates an explicitly separate repetition. The verifier evaluates the event's title, summary, participants, time interval, and existing member memories against the proposed memory. This ensures that mere topical overlap is insufficient for event assignment.

For each newly extracted memory m, HiMem retrieves at most $K_E = 5$ candidate events with a cosine similarity of at least $\delta = 0.80$ and verifies them in descending order of similarity. The first event satisfying $v(m, e) = 1$ is selected. Memories that fail to match any existing candidate are conservatively assigned to newly instantiated event anchors, ensuring that every memory is structurally grounded in the hierarchical graph. Formally, an event node is represented as a tuple $(u_e, t_e, z_e, P_e, [\tau_e^-, \tau_e^+], c_e, \mathbf{h}_e)$, encompassing its unique identifier, title, retrieval summary, participants, time interval, confidence score, and dense embedding. Upon successful assignment, the event's representation is dynamically updated to incorporate the new memory, maintaining the fidelity of the hierarchical structure.

D.2 RELATION CONSTRUCTION

HiMem constructs the hierarchical memory graph through two complementary mechanisms: deterministic online linking and LLM-verified offline semantic relation construction. For online construction, HiMem identifies distinctive phrases across active memories using inverse document frequency (IDF). A phrase x is distinctive when

$$\text{idf}(x) = \log \frac{N}{\text{df}(x)} \geq \max\{1, \tfrac{1}{2} \log N\}. \tag{13}$$

After stop-word filtering and exact phrase-boundary checking, HiMem establishes SHARES-ENTITY edges between cross-event memories containing the same distinctive phrase. If their timestamps differ by at most 30 days, a directed CHRONOLOGICAL edge is added to record their temporal order. Crucially, these online links do not assert event identity; rather, they bridge different verified occurrences through observable entities and temporal dynamics, complementing the occurrence-verified event anchors.

For offline construction, HiMem leverages event anchors as structural context to discover typed semantic relations. For a given anchor event e, HiMem retrieves up to five neighboring events with an event-embedding similarity of at least 0.60. An LLM-based relation judge then evaluates the active memories within the anchor event (up to 30) and the neighboring events (up to 60 in total) to predict the relation type:

$$r_{ij} \in \{\text{COREFERS-TO}, \text{DEVELOPS}, \text{SUPPORTS-INFERENCE}, \text{CONTRADICTS}, \emptyset\}. \tag{14}$$

Here, COREFERS-TO denotes alternative references to the same entity, proposition, or occurrence; DEVELOPS indicates an explicit continuation, realization, consequence, or update; SUPPORTS-INFERENCE joins distinct facts that support a specific intermediate conclusion; and CONTRADICTS preserves incompatible claims about the same attribute within an applicable timeframe. The default prediction is $\emptyset$. An edge is instantiated only if both memory identifiers are valid, at least one endpoint belongs to the anchor event, and the predicted confidence exceeds 0.65.

D.3 PROVENANCE-PRESERVING MEMORY CONSOLIDATION

To ensure the reliability of the hierarchical structure, HiMem performs provenance-preserving memory consolidation strictly within the boundaries of occurrence-verified event anchors. For each event containing between 2 and 50 memories, HiMem orders its members chronologically and retrieves active memories from the three nearest external events (with anchor similarity ≥ 0.70) as reference context. The event summary and its member memories define the primary occurrence, while the neighboring memories are exclusively used to resolve ambiguous entities and temporal updates.

The consolidation model processes each member memory m_j and returns its updated lifecycle status a_j, an optional canonical text $\bar{d}_j$, and the supporting identifiers S_j:

$$(a_j, \bar{d}_j, S_j) = \text{Consolidate}(m_j; \mathcal{M}_e, \mathcal{N}_e), \tag{15}$$

where $\mathcal{M}_e$ and $\mathcal{N}_e$ denote the member and neighboring memories, respectively. A memory is labeled as a *duplicate* only if it is an exact paraphrase lacking independent evidential value, and *superseded* only if a later, more reliable incompatible statement exists; otherwise, it remains *active*. Canonical rewriting resolves pronouns and ambiguities while strictly preserving the original category tag, date, and fine-grained details. Crucially, to maintain the provenance-preserving nature of HiMem, the original memory and its provenance pointer are retained rather than deleted, ensuring that the exact local context can be restored on demand for fine-grained evidence verification. Finally, numeric state summaries are materialized only when they contain a valid date and a confidence score of at least 0.70. This consolidation process precedes offline relation construction, ensuring that semantic edges are proposed from a refined, event-conditioned memory set.

E RETRIEVAL AND ANSWERING DETAILS

E.1 MEMORY-LEVEL CANDIDATE GENERATION

Dense retrieval initially retrieves a pool of $\min(100, \max(30, 3K))$ atomic memory candidates. In parallel, event-mediated retrieval identifies up to $\min(20, \max(3, K))$ event anchors that meet a query similarity threshold of 0.50 and an event confidence threshold of 0.60, then returns the top $\min(8, \max(5, K))$ member memories most similar to the query for each per event. Meanwhile, relational expansion initiates from the top-10 dense retrieval seeds, traverses the graph up to two hops, and retains a maximum of $\min(60, \max(30, 3K))$ graph-based candidates.

Let $p = (m_s, r_1, \ldots, r_h, m)$ be the shortest retained path from dense seed m_s to m. Route support is

$$r(m) = \max \begin{cases} \text{clip}(s_{sem}(e,q)c_e), & m \in e \text{ via an event hit}, \\ \text{clip}\Big(s_{sem}(m_s,q)\pi(r_h)0.75^{h-1}\prod_{r\in p} c_r\Big), & m \text{ reached through } p, \\ 0, & \text{dense-only candidate}, \end{cases} \tag{16}$$

where s_{sem} denotes cosine scores, c_e is event confidence, c_r is edge confidence (defaulting to one for deterministic edges), and clip bounds support to $[0, 1]$. If several routes reach the same memory, HiMem keeps the maximum support and records all route labels. To quantify the reliability of relational paths, we assign terminal-relation priors based on semantic strength: 0.85 for COREFERS-TO; 0.75 for DEVELOPS and event-level coreference/development; 0.70 for BELONGS-TO-EVENT; 0.45 for SHARES-ENTITY and SUPPORTS-INFERENCE; and 0.15 for CHRONOLOGICAL. Unknown relations default to a prior of 0.35. Notably, paths containing CONTRADICTS are preserved as counter-evidence but are assigned zero positive structural support.

E.2 UNIFIED MEMORY SCORING

After union and identifier deduplication, every candidate memory is rescored to ensure a fair comparison across different retrieval routes. The primary relevance is determined by the cosine similarity between the query embedding and the memory's canonical embedding (or its original embedding if no canonical form exists). To integrate structural signals without overwhelming semantic relevance, Equation 4 is instantiated as

$$s_M(m|q) = s_{\text{sem}}(m,q) + 0.15r(m,q) + 0.025I_{\text{agree}}(m,q) + b_{\text{date}}(m,q) - 0.08I_{\text{superseded}}(m), \tag{17}$$

where $I_{\text{agree}}(m,q) = 1$ when a memory is retrieved by both dense search and a structural route (event or graph). The temporal bonus b_{date} awards 0.015 for an exact year match and 0.025 for an exact month-name match with the query, capped at a maximum of 0.05. Exact duplicates are discarded, while superseded memories are retained but penalized by 0.08. The top-K remaining memories form $\widehat{\mathcal{M}}_q$.

E.3 SOURCE-RECORD RANKING

The candidate set for source records (chunks) is constructed as $\mathcal{U}_C(q) = \mathcal{B}(q) \cup \{c_{\rho(m)} : m \in \widehat{\mathcal{M}}_q\}$, where $\mathcal{B}(q)$ contains positive-score BM25 hits and $\rho(m)$ is the provenance pointer linking a memory to its source chunk. BM25 tokenization lowercases the text, removes stop words while preserving numbers, and applies stemming. At most $\min(30, \max(10, 2K))$ positive BM25 candidates are retained from a pool of 2,000 records.

For each record c, we compute four independent signals. Let $f_c = \max_{m:\rho(m)=c} s_M(m \mid q)$ denote the highest fact score pointing to c, and let F_c be its min–max normalization across all pointer-derived candidates. Let $B_c = 1/\log_2(\text{rank}_{\text{BM25}}(c) + 1)$ represent the reciprocal-log BM25 rank, with $B_c = 0$ if c is not a BM25 hit. If Q is the set of non-stopword query terms, the lexical coverage is defined as $L_c = \frac{1}{|Q|}\sum_{w\in Q} \mathbb{I}[w \text{ occurs in } c]$. Finally, $G_c = \max_{m:\rho(m)=c} r(m)$ captures the maximum structural route support among memories linked to c. The implemented source score is:

$$s_C(c \mid q) = 0.22F_c + 0.60B_c + 0.13L_c + 0.05G_c. \tag{18}$$

The highest-scoring C records are returned as the final raw context. To ensure robust lexical recall, a soft lexical-coverage guard is applied: the highest-ranked BM25 record may replace the weakest selected record only if it has positive lexical coverage ($L_c > 0$) and its unified score is at least 80% of the weakest selected score. This design ensures that source retrieval remains independent and that a high BM25 score is not artificially propagated to all memories extracted from that chunk.

E.4 CONTEXT ASSEMBLY AND SUPPLEMENTARY RETRIEVAL

Before constructing the final prompt, HiMem deduplicates atomic memories by their unique identifiers and source records by content to eliminate redundancy. The answer prompt is then assembled with the query q, the top-K ranked atomic memories, and the top-C ranked original source records.

Table 5: Implementation hyperparameters for memory formation, graph evolution, and retrieval.

Parameter	**Value**
Memory Formation	
Minimum attention-layer index	8
Maximum attention-layer index	11
Similarity-sweep initial threshold	0.20
Similarity-sweep increment	0.05
Similarity-sweep maximum threshold	0.50
Temporal boundary	6 hours
Minimum user messages for a non-short fragment	2
Minimum total messages for a non-short fragment	4
Graph Construction and Consolidation	
Event proposal candidate count	5
Minimum event-proposal cosine similarity	0.80
Online temporal window	30 days
Offline event-neighbor count	5
Minimum event-neighbor cosine similarity	0.60
Edge prompt anchor budget	30 memories
Edge prompt total budget	60 memories
Minimum semantic-edge confidence	0.65
Consolidation context size	3
Minimum consolidation-context cosine similarity	0.70
Minimum consolidated-event size	2 memories
Maximum consolidated-event size	50 memories
Retrieval and Ranking	
Memory budget K	20
Source-record budget C	5
Dense-pool minimum	30
Dense-pool multiplier on K	3
Dense-pool maximum	100
Minimum event-retrieval cosine similarity	0.50
Minimum event confidence	0.60
Graph seeds	10
Graph traversal depth	2 hops
Maximum graph-traversal pool	60
Hop decay	0.75
Graph-support weight	0.15
Route agreement bonus	0.025
Superseded penalty	0.08
Fact-support weight F	0.22
BM25-rank weight B	0.60
Lexical-coverage weight L	0.13
Graph-support source weight G	0.05

Notably, structural metadata such as graph paths and event/topic summaries are strictly excluded from the generation context, as they serve as retrieval aids rather than primary evidence.

To address cases where the initial evidence is insufficient, HiMem incorporates a supplementary retrieval fallback. If the first generated answer explicitly reports a lack of evidence, HiMem triggers a supplementary pass. Specifically, it ranks topic summaries by cosine similarity to the query, applies an exact-keyword bonus of $\min(0.03n_{\text{hit}}, 0.15)$, and selects summaries with a final score of at least 0.25. The corresponding original records for these summaries are then loaded to expand the evidence pool. The answer is subsequently regenerated from this enriched context, ensuring that the fallback mechanism remains strictly grounded in the stored interaction text.

E.5 Implementation Hyperparameters

Table 5 lists the operational settings used by the full method. Values are fixed across LoCoMo questions; K and C are varied only in the retrieval-budget experiment.

Table 6: Accuracy (%) on LongMemEval-S using GPT-4.1-mini. Baseline results are reported by EverMemOS (Hu et al., 2026a); HiMem uses the same benchmark and evaluation settings. The best result in each column is in bold.

Method	LongMemEval-S						
	SS-User ↑	SS-Assistant ↑	SS-Pref ↑	Multi-Session ↑	Know. Update ↑	Temporal ↑	Overall ↑
MemU	67.14	19.64	76.67	42.10	41.02	17.29	38.40
Zep	92.90	75.00	53.30	47.40	74.40	54.10	63.80
Mem0	82.86	26.78	90.00	63.15	66.67	72.18	66.40
MemOS	95.71	67.86	**96.67**	70.67	74.26	77.44	77.80
EverMemOS	97.14	85.71	93.33	73.68	**89.74**	77.44	83.00
HiMem	**98.57**	**98.21**	93.33	**81.20**	**89.74**	**82.71**	**88.00**

F Additional Experimental Analyses

F.1 Results on LongMemEval-S

In this subsection, we report the performance of HiMem on the LongMemEval-S benchmark to further evaluate its generalization to extended human–assistant histories. The baseline results in Table 6 are reported by EverMemOS (Hu et al., 2026a); its table in turn attributes the non-EverMemOS baselines to the official MemOS evaluation (Li et al., 2025). These results use the same LongMemEval-S split, GPT-4.1-mini answer model, and official evaluation protocol as our HiMem experiment.

As shown in Table 6, HiMem achieves an overall accuracy of 88.00%, outperforming the strongest reported baseline, EverMemOS, by 5.00 percentage points. Notably, HiMem leads the evaluated systems in five out of the six categories. It demonstrates particularly substantial gains over EverMemOS in single-session assistant recall ($+12.50\%$), multi-session synthesis ($+7.52\%$), and temporal reasoning ($+5.27\%$). While MemOS remains slightly ahead on single-session preference questions, HiMem matches EverMemOS on knowledge update tasks. These results strongly corroborate our findings on the LoCoMo benchmark: the structure-aware multi-source retrieval mechanism is particularly advantageous when the required answer evidence must be seamlessly integrated across different speakers, extended sessions, or complex temporal references.

F.2 Complete Evidence-Support Results

In this section, we report the exact values underlying Figure 4. To evaluate retrieval effectiveness, our evidence-equivalence judge compares retrieved atomic memories against each annotated LoCoMo evidence turn. After excluding questions with unresolvable evidence annotations, the analysis encompasses 1,535 questions and 2,358 evidence instances. Results are micro-averaged over these instances, meaning an evidence turn is counted multiple times if it supports different questions. To isolate the impact of retrieval depth, source records are fixed to those returned by the $(K, C) = (20, 5)$ configuration across all displayed cutoffs, and only the ranked memory list is progressively truncated.

Increasing the memory budget K from 1 to 20 raises atomic-memory support by 41.56 percentage points. Notably, the combined support rises by 10.52 points, as many misses at low K are already compensated by the fixed source records. Crucially, event-only support grows steadily from 8.99% to 18.45%, demonstrating that deeper event-mediated retrieval continuously recovers critical evidence that eludes direct dense matching. Concurrently, graph structural participation rises from 0.04% to 4.20%, while the unsupported share correspondingly drops to 10.73%. This inverse relationship confirms that structural expansion effectively plugs the gaps left by purely semantic retrieval.

F.3 Evidence Support by Question Type

Table 8 presents a detailed category-wise breakdown of evidence support at $K = 20$, utilizing the same attribution methodology and fixed source records as the aggregate analysis.

Table 7: Complete evidence-support results (%). Dense, event-only, raw-context-only, and unsupported form a disjoint partition. Graph structural support (†) is non-exclusive.

K	Dense	Event-only	Graph†	Raw-only	Unsupported	Memory	Combined
1	28.80	8.99	0.04	40.97	21.25	37.79	78.75
3	42.58	12.98	0.08	25.78	18.66	55.56	81.34
5	48.22	14.55	0.34	20.48	16.75	62.77	83.25
10	54.88	16.96	1.78	14.21	13.95	71.84	86.05
15	58.48	17.73	3.31	11.58	12.21	76.21	87.79
20	60.90	18.45	4.20	9.92	10.73	79.35	89.27

Table 8: Evidence support by LoCoMo question type at $K = 20$ (%). N_E is the number of annotated evidence instances; graph support is non-exclusive.

Type	N_E	Dense	Event-only	Graph	Raw-only	Unsupported	Memory	Combined
Multi-hop	881	54.48	16.23	5.90	11.24	18.05	70.72	81.95
Temporal	374	68.45	17.91	1.60	9.63	4.01	86.36	95.99
Open-domain	208	47.12	13.94	4.33	9.13	29.81	61.06	70.19
Single-hop	895	67.26	21.90	3.58	8.94	1.90	89.16	98.10

As expected, temporal and single-hop questions achieve the highest combined support rates, at 95.99% and 98.10%, respectively, due to their relatively straightforward evidence localization. In contrast, multi-hop and open-domain questions remain inherently more challenging, yet they exhibit the highest graph participation rates (5.90% and 4.33%, respectively). This distribution pattern strongly aligns with our design intuition: structural retrieval is disproportionately valuable for compositional evidence that requires cross-session integration. While route frequency alone does not establish a causal accuracy gain, this hypothesis is rigorously validated by the targeted ablation results on typed relations and retrieval components presented in Experiment 3.

F.4 Retrieval-Budget Results

Table 9 reports the exact numerical values underlying Figure 3. We jointly scale both the memory budget (K) and the source-record budget (C) because they capture complementary evidence at distinct granularities; holding either fixed would obscure the distinct contributions of deeper atomic recall versus source-level context reconstruction.

As detailed in Table 9, the most substantial early performance gain occurs when scaling from $(K, C) = (1, 1)$ to $(3, 1)$, where an increment of merely 111 answer tokens yields a significant 5.78-point improvement in accuracy. While accuracy continues to improve with increased retrieval depth, the marginal gains exhibit diminishing returns beyond $K = 10$. Consequently, the $(20, 5)$ configuration is selected for our principal evaluation, as it resides near the knee of the accuracy-token curve, offering an optimal efficiency-accuracy trade-off. Expanding the budget further to $(30, 8)$ incurs a substantial overhead of 2,410 additional tokens for a marginal accuracy gain of only 0.52 points, confirming that our framework achieves peak retrieval efficiency at moderate budgets.

F.5 Category-Wise Ablation Sensitivity

Figure 5 provides a detailed category-wise visualization of the performance changes underlying Table 2. Unlike the aggregate overall accuracy, this fine-grained analysis reveals which specific reasoning abilities are most dependent on each architectural component.

Table 9: HiMem accuracy and answer-input cost across retrieval budgets on LoCoMo.

Memory budget K	Source budget C	Avg. answer tokens	Accuracy (%)
1	1	1,064	76.56
3	1	1,175	82.34
5	2	1,901	87.34
10	3	2,796	91.10
15	4	3,697	91.43
20	5	4,548	92.60
30	8	6,958	93.12

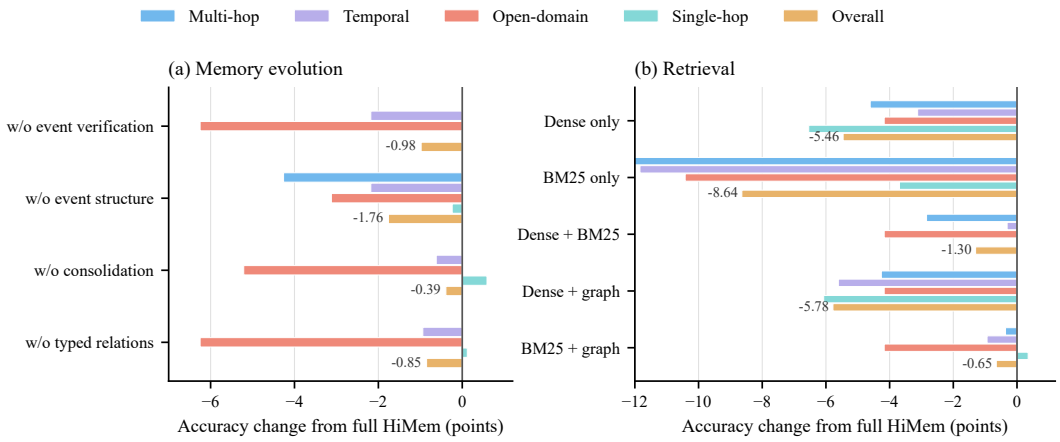

Figure 5: Accuracy change relative to full HiMem for memory-evolution and retrieval ablations. Negative values indicate degradation; numeric labels report the overall change.

For the memory construction components, the impact is most pronounced on open-domain questions. Removing event verification, consolidation, or typed relations degrades performance in this category by 5.21–6.25 points, highlighting their critical role in resolving complex, unconstrained queries. Furthermore, removing the complete event structure causes the largest construction-side drop on multi-hop questions (-4.26 points), which directly corroborates our design intuition that event anchors are essential for linking distributed evidence across different sessions.

The retrieval ablations exhibit a broader and more severe impact across categories. Relying solely on BM25 leads to a substantial 17.02-point drop on multi-hop questions, demonstrating the insufficiency of purely lexical matching for compositional reasoning. Conversely, dense-only retrieval suffers a 6.54-point loss on single-hop questions and a 5.46-point drop overall, underscoring the necessity of structural and lexical expansion. Finally, the consistent performance gap between the full HiMem and the BM25+graph variant confirms that direct semantic candidates (dense retrieval) remain indispensable, providing a crucial baseline of high-precision matches even when structural and source-level evidence are fully leveraged.

F.6 Case Studies: Bridging Semantic and Temporal Gaps

Table 10 presents two representative case studies where HiMem successfully retrieves the necessary distributed evidence and answers correctly, while the strongest baseline, EverMemOS, fails. The first case demonstrates HiMem's capability in cross-session evidence composition via structure-aware graph expansion. The second case highlights its ability to perform source-grounded temporal normalization by retrieving the original context to resolve relative dates.

Table 10: Representative questions answered correctly by HiMem but incorrectly by EverMemOS in the archived paired snapshot. Ranks refer to HiMem's returned memory list.

Field	Evidence and outcome
Question 1	`locomo_0_qa60` (multi-hop): "What instruments does Melanie play?" Gold answer: clarinet and violin.
Gold evidence	`D15:26` states that Melanie plays clarinet; `D2:5` states that playing violin helps her recharge.
HiMem route	The clarinet memory is retrieved directly at rank 16. A one-hop graph expansion retrieves the violin memory at rank 20. HiMem combines the two records and answers "clarinet, violin."
Baseline answer	EverMemOS returns only clarinet, omitting the evidence from the earlier session.
Question 2	`locomo_0_qa80` (temporal): "When did Melanie buy the figurines?" Gold answer: 21 October 2023.
Gold evidence	In `D19:2`, dated 22 October 2023, Melanie says that she bought the figurines "yesterday."
HiMem route	HiMem retrieves the corresponding episodic memory at rank 19 together with its source record, resolves the relative date, and answers "2023-10-21."
Baseline answer	EverMemOS reports that no purchase by Melanie is available and therefore fails to resolve the date.

Table 11: Accuracy (%) across three independent GPT-4.1-mini judge calls on the archived paired snapshot. Mean and standard deviation are computed over the three run-level accuracies; Majority applies a two-of-three vote to each question.

Method	Judge 1	Judge 2	Judge 3	Mean $\pm$ Std.	Majority
EverMemOS	90.45	90.39	90.65	90.50 ± 0.11	90.32
HiMem	**92.47**	**92.21**	**92.14**	$\mathbf{92.27 \pm 0.14}$	**92.21**

F.7 Judge Stability and Paired Significance

Table 11 reports the accuracy across three independent GPT-4.1-mini judge calls for the same frozen HiMem and EverMemOS answers used in the paired analysis. Consequently, the reported standard deviation exclusively captures inter-judge variance, strictly isolating it from the stochasticity of memory construction or answer generation.

HiMem consistently outperforms EverMemOS across all three independent judge calls, achieving accuracies of 92.47%, 92.21%, and 92.14% compared to EverMemOS's 90.45%, 90.39%, and 90.65%, respectively. Crucially, the performance gap between the two methods (approximately 2 percentage points) is substantially larger than the within-method inter-judge standard deviations (0.11 for EverMemOS and 0.14 for HiMem). This indicates that the evaluation noise introduced by the LLM judge is negligible compared to the method-level difference. Under majority-vote aggregation, HiMem maintains a clear advantage with 92.21% accuracy against EverMemOS's 90.32%. These statistical results confirm that the observed performance gain is robust and stable, and is not an artifact of the stochasticity inherent in the LLM-as-a-judge evaluation process.