Schema-Mem: Structured and Evolving Memory for Long-term Conversational Knowledge

Anonymous EMNLP submission

Abstract

Long-term conversational agents require memory systems that continuously acquire and maintain user-specific knowledge from unstructured dialogue histories. Existing LLM memory solutions often rely on flat text storage or coarse summaries, which suffer from redundancy and weak factual grounding, making it difficult to reason over evolving information in long-term interactions. To address this issue, we propose *Schema-Mem*, a structured and evolving memory framework inspired by Schema Theory in cognitive psychology, which represents conversational knowledge as fact-level schema units. Schema-Mem incrementally builds a connected memory graph by extracting key facts from each dialogue turn, linking related historical units through hybrid lexical–semantic search, and refining memory via LLM-based verification to resolve conflicts and incorporate new details. For response generation, it performs multi-source retrieval to provide grounded evidence for reliable answers. Extensive experiments on LoCoMo and LongMemEval show the efficacy of Schema-Mem in maintaining long-term memory and producing reliable responses during extended interactions.

1 Introduction

Large Language Models (LLMs) have become a strong foundation for conversational agents (Matarazzo and Torlone, 2025; Yi et al., 2025), but they still struggle with long dialogue histories due to limited context windows and lost-in-the-middle effects (Baek et al., 2025). Without explicit memory, their ability to perform long-term reasoning and maintain consistency with prior interactions degrades substantially (Jia et al., 2025).

To address these limitations, recent research augments LLM-based conversational agents with explicit memory mechanisms that retain and reuse information beyond the fixed context window (Du, 2026; Zhong et al., 2024; Xu et al., 2025b). Early

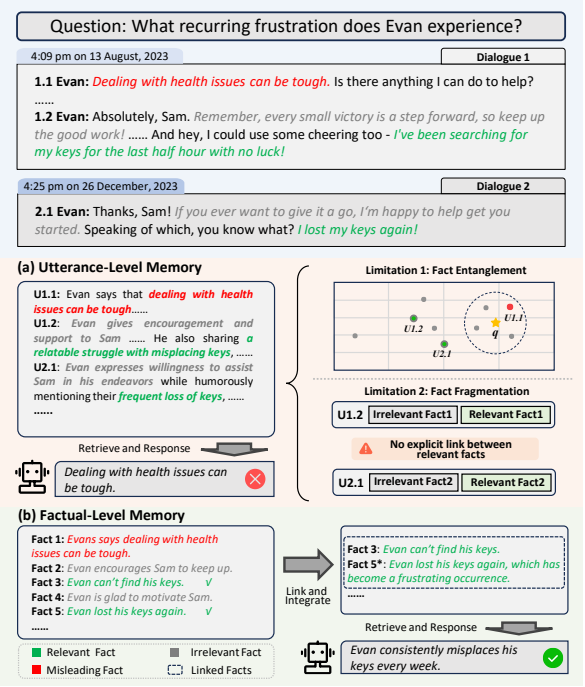


Figure 1: Comparison of memory at different levels.

designs store dialogue histories as discrete retrievable text chunks at different levels of granularity (e.g., utterance, turn, or session level) (Pan et al., 2025; Tan et al., 2025; Yue et al., 2026), but often fail to capture the semantic and relational dependencies in the conversational context. More recent methods impose explicit topological structures over proposition-level memory units, organizing them as trees, graphs, or hierarchical abstractions to preserve relationships and support long-term reasoning (Wu et al., 2025; Zhang et al., 2026; Gutiérrez et al., 2025; Xu et al., 2025a; Hu et al., 2025).

Despite these advances, existing memory representations still struggle to support reliable factual reasoning and long-term maintenance. Current systems mostly store dialogue history as utterance-level text chunks through coarse segmentation or summarization, rather than explicitly representing

the underlying facts. As illustrated in Figure 1(a), this utterance-level design suffers from two key limitations: **fact entanglement**, where facts stored in the same memory unit semantically interfere with one another, making relevant facts harder to retrieve, and **fact fragmentation**, where related facts are scattered across different utterances without explicit connections. Consequently, useful evidence can be both diluted and disconnected during memory access. For example, utterances 1.2 and 2.1 contain clues that Evan repeatedly loses his keys, but these clues are entangled with unrelated supportive content, while across 1.2 and 2.1 the related clues remain fragmented and cannot be naturally combined into a coherent recurring fact. As a result, the memory system fails to surface this evidence effectively and instead relies on the misleading statement "dealing with health issues can be tough," which leads to an incorrect answer. In contrast, Figure 1(b) shows that factual-level memory provides a finer-grained representation, making relevant facts easier to retrieve precisely and to further link and integrate effectively, thus better supporting accurate response generation.

To establish factual-level memory and structured associations among related memories, we propose Schema-Mem, a structured memory framework that stores and evolves conversational knowledge at the factual level. Inspired by Schema Theory in cognitive psychology (Anderson and Pearson, 1984), which posits that human knowledge is organized into structured schemas incrementally refined as new information arrives, Schema-Mem represents salient facts as schema units and links them to related historical units, forming a structured knowledge scaffold for cross-turn aggregation and continual refinement. Specifically, Schema-Mem first extracts key information from conversation histories into schema units that capture factual content and salient elements. It then identifies candidate-related units via a hybrid retrieval strategy combining lexical matching with semantic similarity, and uses an LLM to validate relational relevance and establish reliable links. Schema-Mem further consolidates new units with these candidates to enrich missing details and resolve potential conflicts, allowing memory to evolve with the conversation rather than naively accumulate. At inference time, Schema-Mem retrieves query-relevant schema units together with their linked context, grounding the LLM with both structured facts and supporting dialogue evidence to improve retrieval precision and response consistency.

In summary, the main contributions of our work can be summarized as follows:

- We investigate approaches to enhance the long-term memory capabilities of LLM-based conversational agents, offering a new perspective on designing more effective memory mechanisms for long-horizon dialogue understanding and generation.
- We propose Schema-Mem, a structured and evolving memory framework that stores salient conversational knowledge as fact-level schema units, with continual schema linking and updating to aggregate related facts and maintain memory consistency over time.
- We conduct extensive evaluations of Schema-Mem on LoCoMo and LongMemEval, demonstrating that it consistently outperforms strong baselines in long-context conversational settings. The code is available on `https://anonymous.4open.science/r/Schema-Mem-12D0`

2 Related Work

Memory Systems for LLM-based Agents. LLMs face significant challenges with long-context information in extended dialogs (Liu et al., 2025; Li et al., 2024), due to context window limits and degraded comprehension near the upper bound (Baek et al., 2025). Recent work augments LLMs with external memory modules: MemGPT (Packer et al., 2023) introduces virtual context management that transfers information between LLMs and external storage; MemoryBank (Zhong et al., 2024) draws on human memory decay and reinforcement to update and selectively forget information; A-Mem (Xu et al., 2025b) adopts structured memory evolution with interconnected units to capture changes in long-term dialogues. However, these approaches typically treat memory as unstructured text (Zhang et al., 2024; Zeng et al., 2024), where coarse granularity hinders precise retrieval and updating of fine-grained facts (Zhang et al., 2024; Du et al., 2025), and new information is appended rather than integrated (Xu et al., 2025b; Zhang et al., 2024). In contrast, Schema-Mem structures memory at the proposition level, organizing facts into schema units and explicitly linking related units across dialogues, enabling precise retrieval, incremental integration, and coherent long-term representation.

Structured Knowledge Extractions and Applications. Extracting structured knowledge from unstructured text is a fundamental NLP problem (Dagdelen et al., 2024; Wu et al., 2024b; Mo et al., 2025), traditionally studied through tasks such as named entity recognition, relation extraction, and event extraction (Li et al., 2022; Nguyen and Grishman, 2015; Xiang and Wang, 2019). Recent LLM-based approaches construct richer knowledge structures for retrieval and reasoning, including graph-based retrieval (Guo et al., 2024), structured reasoning chains (Bao et al., 2026), temporal knowledge graphs (Rasmussen et al., 2025), and dynamically composable memory graphs (Ke et al., 2025). However, these methods mainly rely on static or one-shot construction with limited support for continuous memory evolution between conversations. In contrast, Schema-Mem treats proposition-level schema units as persistent memory elements that are dynamically linked and consolidated over time, enabling evolving structured memory for long-term conversational question answering.

3 Methodology

We propose Schema-Mem for implementing long-term conversational memory, aiming to generate consistent responses by storing and utilizing information spanning across historical conversation. Given a historical conversation $U = \{(u_1, \tau_1), (u_2, \tau_2), \ldots, (u_t, \tau_t)\}$, where u_t is the content of the utterance and τ_t its associated timestamp at turn t, the framework aims to retain the conversational history and generate responses to a query q that are consistent with the context of U. As shown in Figure 2, Schema-Mem comprises four main modules: (1) **Schema Extraction**, which extracts raw dialogue into predicate-argument structures and metadata; (2) **Schema Linking**, which establishes semantic connections between newly generated schema units and historical units through hybrid retrieval and LLM-based verification; (3) **Schema Update**, which dynamically integrates new schema units with historical context to ensure memory consistency and evolution; and (4) **Schema Retrieval and Response Generation**, which synthesizes evidence to generate accurate responses.

3.1 Schema Extraction

Schema Extraction aims to identify key facts from raw conversational histories and transform them into a structured memory representation that is storable, retrievable, and capable of evolving. Specifically, Schema-Mem first leverages an LLM to analyze each utterance and extract key semantic information. To preserve contextual coherence across turns, the summary from the previous turn of the utterance s_{t-1} is concatenated with the current content of the utterance u_t and the timestamp τ_t and fed into the LLM:

$$(s_t, K_t, T_t) = \mathrm{LLM}(s_{t-1}, u_t, \tau_t), \qquad (1)$$

where s_t, K_t, and T_t denote the summary, keywords, and categorical tags of the current turn, respectively. Tags provide higher-level semantic categories that support coarse topical organization and memory filtering. u_t, τ_t, s_t, K_t and T_t together compose the utterance information, which is defined as $d_t = \langle u_t, \tau_t, s_t, K_t, T_t \rangle$. d_t is stored in the utterance repository $\mathcal{D}$ for response generation. After obtaining the utterance-level information, Schema-Mem further extracts factual information from the conversation. We refer to such factual information as propositions, which are represented as $p_i^t = \langle c_i^t, v_i^t \rangle$, where c_i^t denotes the content of the proposition, and v_i^t denotes the set of proposition keywords, which explicitly includes the predicate and arguments of the proposition. Concretely, for the current turn, we prompt the LLM to extract factual knowledge and output it in the form of a proposition set:

$$\mathcal{P}_t = \mathrm{LLM}(u_t, \tau_t, u_{t-3:t-1}, \tau_{t-3:t-1}, K_t), \qquad (2)$$

where $\mathcal{P}_t = \{p_1^t, p_2^t, \ldots, p_n^t\}$ is the set of propositions extracted from the current utterance. To ensure completeness of the factual extraction, we additionally provide the preceding three turns $u_{t-3:t-1}$ as a contextual input to the LLM.

Finally, the proposition content, proposition keywords, and the tags of the associated utterance are combined to form a schema unit. The i-th schema unit extracted from u_t is defined as:

$$m_i^t = \langle p_i^t, v_i^t, T_t, \tau_t \rangle. \qquad (3)$$

Schema units serve as the minimal structured storage elements in Schema-Mem. After processing the schema units $\{m_i^t\}_{i=1}^n$ with Schema Linking and Schema Update, they can be stored in the schema repository $\mathcal{M}$.

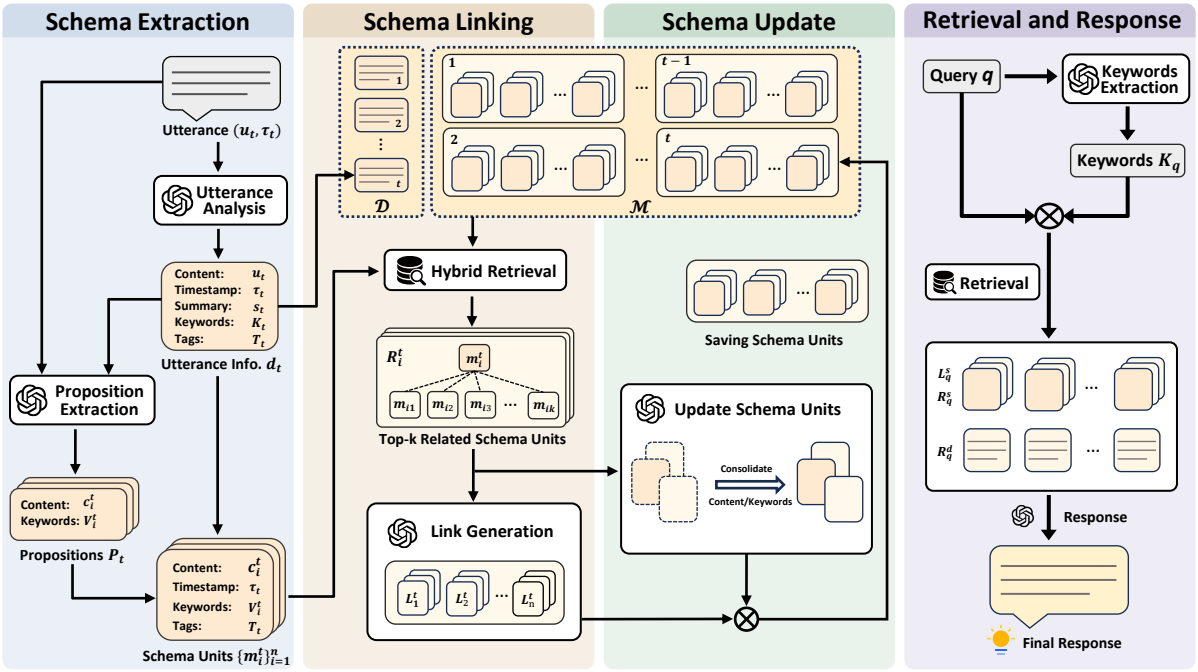


Figure 2: The framework of our Schema-Mem. It consists of four modules: (1) Schema Extraction, (2) Schema Linking, (3) Schema Update, and (4) Schema Retrieval and Response Generation.

3.2 Schema Linking

Schema Linking establishes explicit connections between newly generated schema units and historical schema units, organizing related facts across dialogue turns into a traceable memory structure. To robustly identify cross-turn relevance under paraphrasing and lexical variation, Schema-Mem adopts a hybrid retrieval strategy that combines lexical matching and semantic similarity, followed by an LLM-based refinement step.

Specifically, for the i-th schema unit m_i^t in the current utterance u_t, we retrieve the Top-k most relevant historical schema units from the memory repository. We define a fused relevance score by linearly combining BM25 and embedding-based semantic similarity:

$$\begin{aligned}\mathrm{Score}(m_i^t, m_j) = \alpha \cdot \mathrm{Sim}(m_i^t, m_j) \\ + (1-\alpha) \cdot \mathrm{BM25}(m_i^t, m_j),\end{aligned} \tag{4}$$

where $\mathrm{BM25}(\cdot)$ measures lexical overlap, $\mathrm{Sim}(\cdot)$ denotes semantic similarity (e.g., cosine similarity), and $\alpha \in [0, 1]$ is a weighting factor. Based on this score, we obtain the Top-k candidates:

$$R_i^t = \mathrm{TopK}_{m_j \in \mathcal{M}_{<t}} \mathrm{Score}(m_i^t, m_j), \tag{5}$$

where $R_i^t = \{m_{i1}, m_{i2}, \ldots, m_{ik}\}$ and $\mathcal{M}_{<t}$ denotes the set of all schema units accumulated from dialogue turns before t.

To improve efficiency, instead of invoking the LLM independently for each schema unit, we batch all schema units extracted at the current turn together with their retrieved candidates and perform joint filtering in a single call. Let $\{m_i^t\}_{i=1}^n$ denote the schema unit collection generated at turn t, and $\{R_i^t\}_{i=1}^n$ denote their corresponding Top-k candidate sets. The LLM outputs the final linked set for each schema unit:

$$\{L_i^t\}_{i=1}^n = \mathrm{LLM}\big(\{m_i^t\}_{i=1}^n, \{R_i^t\}_{i=1}^n\big), \tag{6}$$

where $L_i^t \subseteq R_i^t$. The resulting links between m_i^t and elements in L_i^t are used to support subsequent schema updates and retrieval.

3.3 Schema Update

Schema Update aims to fuse and refine the schema units generated at the current turn by integrating relevant information from historical memory. Because conversational facts may evolve across turns, directly appending new units can accumulate redundant or conflicting fragments. Schema-Mem therefore consolidates each current schema unit with its Top-k retrieved historical candidates, updating factual content and keywords to form a more coherent memory representation.

To improve efficiency, Schema-Mem performs schema updating in a batched manner. Let $\{m_i^t\}_{i=1}^n$

denote the set of schema units extracted in the current utterance u_t. For each m_i^t, we retrieve a Top-k set of relevant historical candidate schema units, denoted as R_i^t. We then jointly input $\{m_i^t\}_{i=1}^n$ and their candidate sets $\{R_i^t\}_{i=1}^n$ into the LLM to refine their content and keywords:

$$\mathcal{E}_t = \mathrm{LLM}\big(\{m_i^t\}_{i=1}^n,\ \{R_i^t\}_{i=1}^n\big), \tag{7}$$

where $\mathcal{E}_t$ denotes the set of all schema units updated at turn t, including both current schema units and historical candidate units refined during consolidation. This batched update strategy improves efficiency and promotes memory consistency, and the updated units are stored for subsequent retrieval and response generation.

3.4 Schema Retrieval and Response Generation

Given the user's query, Schema-Mem retrieves the most relevant evidence from the structured memory repository. Specifically, given a query q, we first decompose the query into a set of keywords K_q using LLMs, explicitly capturing the entities, relations, and constraints of the query:

$$K_q = \mathrm{LLM}(q). \tag{8}$$

Subsequently, Schema-Mem employs a hybrid retrieval strategy within the schema unit memory repository, combining BM25 and semantic similarity to retrieve the Top-k schema units most relevant to the query, denoted as:

$$\begin{aligned} R_q^s = \mathrm{TopK}_{m_j \in \mathcal{M}}\Big(&\alpha \cdot \mathrm{Sim}(q, m_j) \\ &+ (1-\alpha) \cdot \mathrm{BM25}(K_q, m_j)\Big), \end{aligned} \tag{9}$$

where m_j represents a schema unit in the schema repository $\mathcal{M}$. At the same time, the linked schema units of R_q^s, denoted as L_q^s, are also extract from $\mathcal{M}$. To compensate for any potential missing contextual details in the structured representation, Schema-Mem also performs semantic retrieval directly from the original dialogue history, retrieving the Top-k most semantically similar utterance segments to the query, denoted as:

$$R_q^d = \mathrm{TopK}_{d \in \mathcal{D}}\big(\mathrm{Sim}(q, d)\big), \tag{10}$$

where d denotes an utterance entry in the utterance repository $\mathcal{D}$. Finally, Schema-Mem combines the retrieved schema units, their linked units, and the retrieved utterance segments as evidence, and feeds them to the LLM together with the query q to generate the final response.

4 Experiments

4.1 Experiment Setup

Datasets and Evaluation Metrics. We evaluate Schema-Mem on two long-term conversational memory benchmarks: **LoCoMo** (Maharana et al., 2024) and **LongMemEval** (Wu et al., 2024a). LoCoMo focuses on long multi-session dialogues with diverse reasoning tasks, while LongMemEval emphasizes dynamic memory capabilities such as information extraction, multi-session reasoning, and knowledge updates. More details about the datasets are provided in Appendix A.

Following established protocols, we report F1 and BLEU-1 (BLEU) on both benchmarks, where F1 measures factual consistency and BLEU evaluates surface-level linguistic similarity between generated responses and ground truth. Following previous work (Fang et al., 2025), for LongMemEval we additionally adopt an LLM-based accuracy metric, using GPT-4o to evaluate prediction correctness with the prompt from (Fang et al., 2025).

Implementation Details. To evaluate the versatility and robustness of Schema-Mem across varying model scales and access paradigms, we adopt two different backbone LLMs. We select GPT-4o-mini (OpenAI, 2024) as the representative for state-of-the-art proprietary models. For open-source experiments, we use Qwen2.5-3B (Yang et al., 2024) with local deployment via SGLang for efficient inference. In both the Schema Linking and Schema Update stages, the retrieval size parameter k is set to 5. During the response generation stage, the retrieval size parameter k is set to 15 for GPT-4o-mini and 10 for Qwen2.5-3B, respectively. The weighting factor α is set to 0.7. We implement the all-minilm-l6-v2 model. For text embedding. All experiments are conducted on a Linux server with two NVIDIA 4090 GPUs.

Baselines. To evaluate the effectiveness of our Schema-Mem, we compare it with a diverse set of state-of-the-art baselines: **LoCoMo** (Maharana et al., 2024), **READAGENT** (Lee et al., 2024), **MemoryBank** (Zhong et al., 2024), **MemGPT** (Packer et al., 2023), and **A-Mem** (Xu et al., 2025b). These methods cover representative long-term memory paradigms, including direct long-context prompting, gist-based reading memory, decay-based memory updating, virtual context management, and agentic structured memory. More details about the baselines are provided in Appendix B.

Table 1: Main results on the LoCoMo benchmark across multi-hop, temporal, open-domain, and single-hop. Best and second-best results are highlighted in **bold** and underlined, respectively.

Model	Method	Category								Avg. Rank	
		Multi Hop		Temporal		Open Domain		Single Hop			
		F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU
GPT-4o-mini	LoCoMo	25.02	19.75	18.41	14.77	12.04	11.16	40.36	29.05	3.75	3.50
	READAGENT	9.15	6.48	12.60	8.87	5.31	5.12	9.67	7.66	5.25	5.25
	MemoryBank	5.00	4.77	9.68	6.99	5.56	5.94	6.61	5.16	5.75	5.75
	MemGPT	26.65	17.72	25.52	19.44	9.15	7.44	41.04	34.34	3.25	3.50
	A-Mem	27.02	20.09	45.85	36.67	12.14	12.00	44.65	37.06	2.00	2.00
	Ours	**30.78**	**23.51**	**51.04**	**41.94**	**15.09**	**13.04**	**49.49**	**42.00**	**1.00**	**1.00**
Qwen2.5-3B	LoCoMo	4.61	4.29	3.11	2.71	4.55	5.97	7.03	5.69	4.25	4.25
	READAGENT	2.47	1.78	3.01	3.01	5.57	5.22	3.25	2.51	5.25	5.25
	MemoryBank	3.60	3.39	1.72	1.97	6.63	6.58	4.11	3.52	5.00	5.00
	MemGPT	5.07	4.31	2.94	2.95	7.04	7.10	7.26	5.52	3.50	3.50
	A-Mem	12.57	9.01	**27.59**	**25.07**	7.12	7.28	17.23	13.12	1.75	1.75
	Ours	**16.17**	**10.33**	23.76	21.44	**11.85**	**10.03**	**25.91**	**20.15**	**1.25**	**1.25**

Table 2: Main results on LongMemEval with different backbone models. Bold numbers indicate the best result within each backbone.

Model	Methods	F1	BLEU	Acc.
GPT-4o-mini	LoCoMo	16.44	12.41	0.16
	MemoryBank	14.69	8.98	0.51
	A-mem	14.84	9.52	0.58
	Ours	**20.94**	**13.17**	**0.66**
Qwen2.5-3B	LoCoMo	11.50	8.25	0.14
	MemoryBank	4.27	2.41	0.13
	A-mem	7.56	4.58	0.14
	Ours	**13.96**	**10.16**	**0.16**

4.2 Experiment Results

The experimental results on LoCoMo and LongMemEval demonstrate the effectiveness and robustness of our approach across diverse reasoning tasks and evaluation settings. As shown in Table 1, our method consistently achieves the best average rank across all task categories under both GPT-4o-mini and Qwen2.5-3B backbones, indicating that the observed improvements are stable across different model architectures. The gains are particularly notable on multi-hop, temporal, and adversarial tasks, which require maintaining long-range dependencies and integrating information across multiple reasoning steps, suggesting that our approach is well-suited to complex reasoning scenarios. Compared with strong memory-based baselines such as A-Mem and MemGPT, our method not only attains higher F1 and BLEU scores but also shows more consistent performance across task categories, especially under adversarial settings. Moreover, the simultaneous improvements in F1 and BLEU indicate that the performance gains reflect improved generation quality rather than metric-specific optimization. On LongMemEval, Table 2 shows that our approach outperforms all baselines across F1, BLEU, and accuracy metrics under different backbone models, demonstrating its effectiveness in retaining and utilizing long-term contextual information. Taken together, the results in Table 1 and Table 2 confirm that our method generalizes well across benchmarks with different evaluation focuses, consistently improving both reasoning accuracy and long-context understanding.

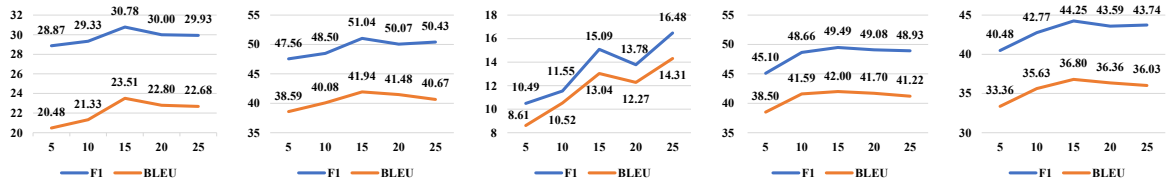
4.3 Ablation Study

Table 3 presents the ablation results on LoCoMo and LongMemEval to evaluate the effectiveness of schema units as well as the linking and updating mechanisms for long-term memory modeling.

Impact of schema units. Removing schema extraction together with linking and updating (w/o schema) causes the largest performance drop, showing that schema units are essential for organizing long-term memories. In the w/o schema extraction setting, schema units are replaced with utterance summaries and keywords while retaining linking

Table 3: Ablation study on LoCoMo and LongMemEval.

Method	Locomo (F1)					Longmemeval	
	Multi Hop	Temporal	Open Domain	Single Hop	Total	F1	Acc.
Schema-Mem	**30.78**	**51.04**	**15.09**	**49.49**	**44.24**	**20.93**	**0.66**
w/o schema	26.55	48.57	12.09	46.10	40.91	16.48	0.40
w/o schema extraction	28.03	50.42	13.92	47.65	42.53	17.39	0.48
w/o schema linking	29.27	49.48	13.70	49.91	43.78	20.19	0.61
w/o linking and update	30.36	50.89	12.24	48.21	43.26	20.07	0.54
w/o link verification	28.74	47.81	12.06	49.00	42.73	20.17	0.60



(a) Multi Hop (b) Temporal (c) Open Domain (d) Single Hop (e) Average

Figure 3: Impact of memory retrieval parameter k across different task categories with GPT-4o-mini as the base model on LoCoMo benchmark.

and update operations. Although this setting partially recovers performance compared with w/o schema, it still performs worse than the full model, indicating that schema units provide a more accurate and stable representation for memory construction and retrieval.

Effectiveness of linking and update. Removing schema linking (w/o schema linking) or disabling both linking and update (w/o linking & update) consistently degrades performance, demonstrating that dynamic schema association and maintenance are crucial for long-term memory reasoning. Moreover, removing the LLM-based link verification process (w/o link verification), where new schema units are directly linked to retrieved candidates without semantic verification, leads to even larger performance degradation than removing linking or updating modules, highlighting the necessity of LLM-guided semantic linking for constructing reliable memory structures.

4.4 Hyperparameter Analysis

We analyze the sensitivity of Schema-Mem to the retrieval size parameter k on LoCoMo using GPT-4o-mini, reporting results for Multi-hop, Temporal, Open-domain, Single-hop, and the overall average. As shown in Figure 3, increasing k from 5 to 15 consistently improves both F1 and BLEU, indicating better evidence coverage, whereas further increasing k to 20 or 25 yields diminishing returns or slight degradation due to redundant and noisy candidates. The drop is more evident in BLEU, suggesting a trade-off between recall and generation precision. Notably, Open-domain tasks continue to benefit from larger k, likely because relevant evidence is sparser and harder to retrieve. Overall, a moderate retrieval size (e.g., $k \approx 15$) offers the best balance in most settings.

4.5 Efficiency Analysis

To evaluate Schema-Mem's efficiency, we compare its computational cost with A-Mem on the LoCoMo dataset using GPT-4o-mini, measuring performance in both the Memorizing and Response phases. In the Memorizing Phase, we measure the average token consumption, number of LLM calls, and runtime required to memorize per utterance. As shown in Table 4, Schema-Mem incurs higher memorization cost than A-Mem, with increased token usage (3061.30 vs. 1953.99), LLM calls (2.71 vs 2.00) and runtime (22.48 s vs. 11.25 s). This additional overhead mainly comes from the structured memory construction process, particularly the extraction of schema units and the subsequent schema linking and update operations. Since memory construction can be performed asynchronously or incrementally in practical systems, the additional

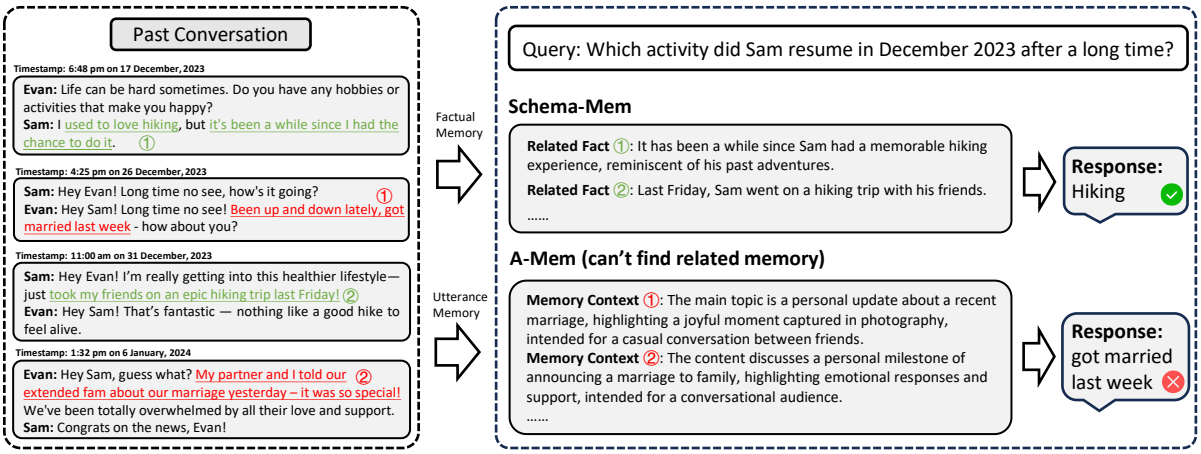


Figure 4: Case study of Schema-Mem.

Table 4: Comparison of token usage, LLM calls, and runtime in memorizing and retrieval phases.

Evaluation	Memorizing		Response	
	A-Mem	Ours	A-Mem	Ours
Token	1953.99	3061.30	10341.29	4584.85
LLM Calls	2.00	2.71	2.00	2.00
Time (s)	11.25	22.48	3.18	2.75

memorization overhead does not necessarily affect user-facing response latency. In the Response Phase, we report the same metrics per query. In contrast, Schema-Mem significantly reduces retrieval and inference cost, consuming less than half the tokens of A-Mem (4584.85 vs. 10341.29) with less runtime (2.75 s vs. 3.18 s). Since Schema-Mem stores fine-grained factual units and explicitly links related memories, the retrieval process can more accurately identify relevant evidence while avoiding large amounts of irrelevant conversational context. As a result, fewer memory entries need to be recalled and provided to the LLM during response generation, reducing both retrieval redundancy and inference overhead.

4.6 Case Study

As shown in Figure 4, we present a representative example where Schema-Mem successfully retrieves the correct evidence from long-term conversation history and produces the correct answer, while A-Mem fails due to irrelevant memory retrieval. Specifically, the query asks: *"Which activity did Sam resume in December 2023 after a long time?"* Given this query, Schema-Mem retrieves proposition-level factual memories that directly capture both the long-term inactivity and the recent resumption, such as *"it has been a while since Sam had a memorable hiking experience"* and *"Sam went on a hiking trip with his friends"*, which provides sufficient and consistent evidence for answering the question correctly (*Hiking*). In contrast, A-Mem retrieves utterance-level memory contexts centered on an unrelated topic (Evan's marriage announcement and family reactions), which do not contain the queried activity and therefore mislead the model into producing an incorrect response (*got married last week*). The contrast shows that structured schema units are more robust to topic drift and better ground answers in relevant evidence, while utterance-level memory introduces off-topic contexts that hurt retrieval and accuracy.

5 Conclusion

In this paper, we propose Schema-Mem, a structured and evolving memory framework that overcomes the limitations of existing conversational memory systems in fact-level storage and memory evolution. To effectively capture factual knowledge from conversations, Schema-Mem analyzes utterance and extracts propositions that represent key facts. These fact-centric representations are stored as schema units, enabling compact and structured long-term memory. Moreover, newly generated schema units are linked with previously stored units and further integrated through knowledge consolidation, allowing the memory to be continuously refined and evolved as the conversation progresses. Extensive experiments on two challenging benchmark datasets demonstrate that Schema-Mem consistently achieves state-of-the-art performance compared to strong baseline methods.

Limitations

While Schema-Mem demonstrates strong performance in two challenging long-term conversational memory benchmarks, several limitations point to promising directions for future improvement.

First, as shown in our efficiency analysis, the structured extraction, linking, and update processes introduce a higher computational cost during the memorizing phase compared to unstructured or lightweight memory baselines. This additional cost arises from the use of LLM-based schema construction and verification to ensure memory quality and consistency. Future work could explore more lightweight extraction modules, partial or selective schema updates, as well as asynchronous or delayed memory construction strategies, with the goal of reducing latency while preserving retrieval accuracy and long-term memory reliability.

Second, Schema-Mem represents conversational knowledge primarily at the factual level. While this design enables precise fact tracking, consolidation, and retrieval across long-term conversation interaction histories, higher-level abstractions, such as long-term user preferences, behavioral patterns, or latent user traits, which are not explicitly modeled. In the future, extending the framework to support hierarchical or multi-level schema representations may enable more abstract understanding, reasoning and improve generalization in complex long-term conversational scenarios.

Overall, these limitations reflect intentional design choices that prioritize memory precision and robustness. Addressing them offers opportunities to further enhance the efficiency, scalability and expressiveness of structured long-term memory for LLM-based conversational agents.

References

Richard C Anderson and P David Pearson. 1984. A schema-theoretic view of basic processes in reading comprehension. *Center for the Study of Reading Technical Report; no. 306*.

Jinheon Baek, Sun Jae Lee, Prakhar Gupta, Geunseob Oh, Siddharth Dalmia, and Prateek Kolhar. 2025. Revisiting in-context learning with long context language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26950–26966, Vienna, Austria. Association for Computational Linguistics.

Meikai Bao, Kai Zhang, Xukai Liu, Qi Liu, Hongke Zhao, and Enhong Chen. 2026. Learn to understand: Knowledge exemplification via multi-agent cooperation for science question answering. *IEEE Transactions on Knowledge and Data Engineering*.

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

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature communications*, 15(1):1418.

Pengfei Du. 2026. Memory for autonomous llm agents:mechanisms, evaluation, and emerging frontiers.

Xingbo Du, Loka Li, Duzhen Zhang, and Le Song. 2025. Memr3: Memory retrieval via reflective reasoning for llm agents. *ArXiv*, abs/2512.20237.

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. 2025. Lightmem: Lightweight and efficient memory-augmented generation. *ArXiv*, abs/2510.18866.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, et al. 2025. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*.

Zixi Jia, Qinghua Liu, Hexiao Li, Yuyan Chen, and Jiqiang Liu. 2025. Evaluating the long-term memory of large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19759–19777.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25961–25970, Suzhou, China. Association for Computational Linguistics.

Cai Ke, Yiming Du, Bin Liang, Yifan Xiang, Lin Gui, Zhongyang Li, Baojun Wang, Yue Yu, Hui Wang, Kam-Fai Wong, and Ruifeng Xu. 2025. Flexibly utilize memory for long-term conversation via a fragment-then-compose framework. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21119–21136, Suzhou, China. Association for Computational Linguistics.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. In *Proceedings of the 41st International Conference on Machine Learning*, pages 26396–26415.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2022. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, et al. 2025. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, Bangkok, Thailand. Association for Computational Linguistics.

Andrea Matarazzo and Riccardo Torlone. 2025. A survey on large language models with some insights on their capabilities and limitations. *arXiv preprint arXiv:2501.04040*.

Belinda Mo, Kyssen Yu, Joshua Kazdan, Joan Cabezas, Proud Mpala, Lisa Yu, Chris Cundy, Charilaos Kanatsoulis, and Sanmi Koyejo. 2025. Kggen: Extracting knowledge graphs from plain text with language models. *arXiv preprint arXiv:2502.09956*.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st workshop on vector space modeling for natural language processing*, pages 39–48.

OpenAI. 2024. Gpt-4o mini model. `https://platform.openai.com/docs/models/gpt-4o-mini`. OpenAI API Documentation, accessed 2026.

Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. 2023. Memgpt: Towards llms as operating systems.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H Vicky Zhao, Lili Qiu, et al. 2025. On memory construction and retrieval for personalized conversational agents. *arXiv preprint arXiv:2502.05589*.

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

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, et al. 2025. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8416–8439.

Yu Wang and Xi Chen. 2025. Mirix: Multi-agent memory system for llm-based agents. *ArXiv*, abs/2507.07957.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2024a. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*.

Haolun Wu, Ye Yuan, Liana Mikaelyan, Alexander Meulemans, Xue Liu, James Hensman, and Bhaskar Mitra. 2024b. Learning to extract structured entities using language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6817–6834, Miami, Florida, USA. Association for Computational Linguistics.

Yaxiong Wu, Yongyue Zhang, Sheng Liang, and Yong Liu. 2025. Sgmem: Sentence graph memory for long-term conversational agents. *arXiv preprint arXiv:2509.21212*.

Wei Xiang and Bang Wang. 2019. A survey of event extraction from text. *IEEE Access*, 7:173111–173137.

Mufan Xu, Gewen Liang, Kehai Chen, Wei Wang, Xun Zhou, Muyun Yang, Tiejun Zhao, and Min Zhang. 2025a. Memory-augmented query reconstruction for llm-based knowledge graph reasoning. *arXiv preprint arXiv:2503.05193*.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025b. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. 2024. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.

Zihao Yi, Jiarui Ouyang, Zhe Xu, Yuwen Liu, Tianhao Liao, Haohao Luo, and Ying Shen. 2025. A survey on recent advances in llm-based multi-turn dialogue systems. *ACM Computing Surveys*, 58(6):1–38.

Juwei Yue, Chuanrui Hu, Jiawei Sheng, Zuyi Zhou, Wenyuan Zhang, Tingwen Liu, Li Guo, and Yafeng Deng. 2026. Hypermem: Hypergraph memory for long-term conversations.

Ruihong Zeng, Jinyuan Fang, Siwei Liu, and Zaiqiao Meng. 2024. On the structural memory of llm agents. *ArXiv*, abs/2412.15266.

Ningning Zhang, Xingxing Yang, Zhizhong Tan, Weiping Deng, and Wenyong Wang. 2026. Himem: Hierarchical long-term memory for llm long-horizon agents. *ArXiv*, abs/2601.06377.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43:1 – 47.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

A Datasets

We conduct experiments on two established datasets for long-term conversational memory.

- **LoCoMo:** LoCoMo is a dataset designed to evaluate very long-term conversational memory (Maharana et al., 2024). Each conversation in LoCoMo has approximately 600 dialogues with 26k tokens and 200 questions on average. Constructed through a human-machine pipeline grounded in specific personas and temporal event graphs, it ensures long-range consistency and causal logic. We utilize LoCoMo to assess Schema-Mem's performance across diverse reasoning tasks, including Single-Hop, Multi-Hop, Temporal Reasoning and Open-Domain Knowledge, testing the system's ability to retrieve distinct facts over extended interaction periods. Following previous work (Kang et al., 2025; Chhikara et al., 2025; Wang and Chen, 2025; Du et al., 2025), we exclude the Adversarial task, which is used to test whether unanswerable questions can be identified.

- **LongMemEval:** LongMemEval is a benchmark specifically designed to test the dynamic capabilities of memory systems in sustained interactions (Wu et al., 2024a). Unlike static benchmarks, it focuses on core long-term memory abilities such as Information Extraction, Multi-Session Reasoning, and notably, Knowledge Updates and Abstention. The benchmark's design involves "needle-in-a-haystack" retrieval across scalable chat histories, providing a robust testbed for our structured graph-based retrieval approach. Following previous work (Ke et al., 2025), we randomly sample 100 episodes from the dataset, where each episode is configured to contain five sessions.

B Baselines

We compare Schema-Mem with five representative baselines, described as follows.

- **LoCoMo** (Maharana et al., 2024): This baseline adopts a direct question-answering paradigm without explicit memory mechanisms, which incorporates the context of the conversation and the questions in the prompt.

- **ReadAgent** (Lee et al., 2024): ReadAgent processes long context documents using a three step framework that segments content, distills concise memory representations, and retrieves relevant information on demand.

- **MemoryBank** (Zhong et al., 2024): By simulating the Ebbinghaus Forgetting Curve, MemoryBank introduces dynamic updates based on time decay, but this scalar mechanism risks prematurely discarding low-frequency critical information.

- **MemGPT** (Packer et al., 2023): MemGPT manages a virtual context window through OS-like function calls to archival storage, prioritizing context management over the automated structural extraction required for deep semantic alignment.

- **A-Mem** (Xu et al., 2025b): Organizing memory as interconnected notes inspired by Zettelkasten, A-Mem supports evolution but operates at a coarser node-level granularity.

C Intermediate-Module Effectiveness Analysis

To further evaluate the effectiveness of the intermediate memory components, we conduct a detailed analysis of the three major modules in the framework: **Schema Extraction**, **Schema Linking**, and **Schema Update**. We assess each stage

using a combination of semantic similarity metrics, coverage measures, and LLM-based evaluations to examine both semantic fidelity and structural consistency throughout the pipeline. We use GPT-4o as the LLM evaluator.

Schema Extraction. The Schema Extraction module aims to derive structured schema units from the original utterances while preserving factual information and minimizing unsupported content. Table 5 summarizes the evaluation results.

Table 5: Extraction stage evaluation results.

Metric	Value
SBERT similarity	0.54
Entity coverage	0.32
LLM faithfulness (1–5)	3.71
Hallucination rate	0.38

SBERT similarity measures the semantic consistency between the extracted content of schema units and the source utterance. Entity coverage reflects the fraction of schema keywords grounded in the original text. LLM faithfulness evaluates factual correctness, while hallucination rate measures the proportion of unsupported content generated during extraction. Overall, the results indicate moderate semantic fidelity, with some inevitable information loss but relatively limited hallucinations.

Schema Linking. The Schema Linking module is responsible for identifying meaningful semantic connections between newly generated schema units and historical schema units. Table 6 presents the evaluation results.

Table 6: Linking stage evaluation results.

Metric	Value
Linked similarity	0.71
Random baseline	0.16
Link Correctness	0.94
Relevance (1–5)	4.22

Linked similarity measures semantic similarity between linked schema pairs, while the random baseline provides a reference score computed from randomly paired schemas. Link Correctness and Relevance are evaluated by the LLM evaluator, measuring the fraction of correct links and the semantic relatedness of linked pairs, respectively. The substantial gap between linked similarity and the random baseline demonstrates that the linking stage reliably captures meaningful semantic relationships. Moreover, the high Link Correctness score and Relevance score further show that the most generated links are factually appropriate and semantically relevant.

Schema Update. The Schema Update module integrates newly acquired information into existing schema units while attempting to preserve previously encoded semantics. Table 7 presents the evaluation results.

Table 7: Updating stage evaluation results.

Metric	Value
Semantic drift	0.72
Update Correctness	0.75

Semantic drift measures the similarity between schemas before and after the update process, where higher values indicate better preservation of original semantics. Update Correctness measures the proportion of schema updates judged by the LLM evaluator as appropriate and factually correct. The results suggest that the updating mechanism maintains strong semantic consistency while still enabling effective integration of new information into the schema representation.

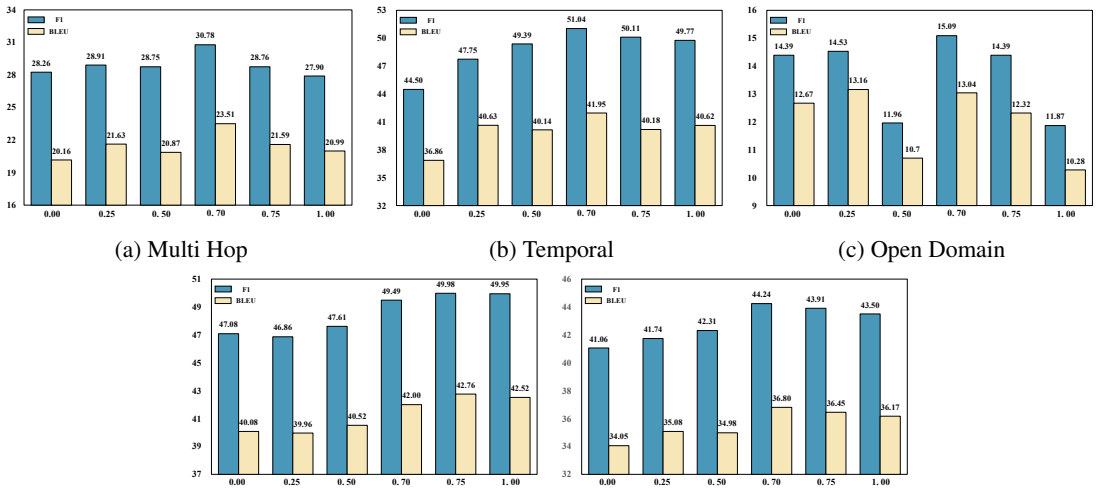
D Analysis of LongMemEval Sampling Variance

Since the original LongMemEval benchmark is relatively large-scale and computationally expensive to evaluate repeatedly, we conduct experiments on a sampled subset of the dataset. To assess robustness, we repeat the evaluation with three different random seeds and report the mean and standard deviation in Table 8.

Table 8: Performance comparison on LongMemEval over three random seeds (mean $\pm$ standard deviation).

Metric	Ours	A-mem
F1	21.25 ± 1.63	14.30 ± 0.38
BLEU	13.67 ± 1.20	8.84 ± 0.49
Accuracy	0.65 ± 0.04	0.57 ± 0.02

The relatively small standard deviations across different runs demonstrate the statistical stability and robustness of our method. Moreover, our approach consistently outperforms the A-mem baseline across all evaluation metrics.



(a) Multi Hop (b) Temporal (c) Open Domain

(d) Single Hop (e) Average

Figure 5: Impact of hybrid retrieval weight α across different task categories with GPT-4o-mini as the base model on LoCoMo benchmark. $\alpha = 0.70$ is the default setting used in our main experiments.

E Hybrid Retrieval Analysis

We further analyze the effect of the hybrid retrieval weight α, which balances lexical matching and embedding-based semantic similarity, with $\alpha \in [0, 1]$, where $\alpha = 0$ denotes BM25-only retrieval, $\alpha = 1$ denotes embedding-only retrieval, and $\alpha = 0.70$ is the default setting used in our main experiments. As shown in Figure 5, hybrid retrieval consistently outperforms either single-source retrieval strategy, confirming that lexical and semantic signals are complementary for long-term conversational memory retrieval. As α increases from 0 to 0.70, the overall F1 improves from 41.06 to 44.24, suggesting that semantic similarity helps recover relevant schema units under paraphrasing and lexical variation. However, embedding-only retrieval remains suboptimal, as $\alpha = 1$ leads to performance degradation on both Open-Domain and Multi-Hop questions, indicating that dense similarity alone is insufficient to preserve exact entity overlap and explicit cross-turn clues required for sparse fact matching and multi-step reasoning. Overall, the results show that a moderately semantic-oriented hybrid weighting yields the most robust performance across task types.

F Prompt Templates

In this section, we present the prompts used in Schema-Mem. The prompts for Schema Extraction, Schema Linking, and Schema Update are shown in Table 9 and Table 10.

Utterance Analysis

Generate a structured analysis of the following content by:
1. Identifying and extracting keywords, including nouns, noun phrases, and phrases containing numerical values or time information.
2. Generating a context summary that synthesizes the "Prior Context" with the current "Content for Analysis"
3. Creating relevant categorical tags.
Format the response as a JSON object:

```
{
  "keywords": [
    // Extract all nouns, noun phrases, and phrases containing numbers (e.g., "50%", "3 items") or time (e.g., "10 AM", "last year").
    // Extract ONLY from "Content for Analysis"
  ],
  "context":
    // one sentence summarizing that clearly states what the speaker means in the Content for Analysis
  "tags": [
    // several broad categories/themes for classification
    // Include domain, format, and type tags
    // At least three tags, but don't be too redundant.
  ]
}
```

Proposition Extraction

You are an expert Information Extraction system.
Task:
Analyze the [Dialogue] and the provided [Entity List].
Break down the information involving these entities into distinct Propositions (factual statements).
Use the optional [Context] section to better interpret the meaning or reference of the current dialogue.
Constraints:
1. **Coverage**: Try to ensure every entity in the [Entity List] appears in at least one proposition, provided there is factual context in the dialogue.
2. **Grouping**: For each proposition, list strictly the entities from the [Entity List] that participate in that specific fact.
3. **Independence**: Each proposition should describe a single event, relationship, or attribute.
4. **Context Use**: If the [Dialogue] contains references (e.g., pronouns or ellipsis) or implicit facts that can only be resolved using [Context], use that context to interpret and generate accurate propositions.
5. **Context Preference**: In case of ambiguity in the [Dialogue], prefer interpretations that are coherent with the [Context], if provided.
6. **Entity Explanation**: If any entity (or its referents, such as pronouns or descriptive phrases) appears multiple times across the [Context] and [Dialogue], generate a separate explanatory proposition for such entity that explains its meaning, role, or relevance based on information from both sources.
[Context]:
{context}
[Dialogue]:
{dialogue}
[Entity List]:
{keywords}

Table 9: Prompts of Schema Extraction.

You are an AI memory evolution agent responsible for managing and evolving a knowledge base.
Analyze MULTIPLE new memory according to content and keywords, also with their several nearest neighbors memory.
Make batch decisions about their evolution.
For each new memory, determine:
1. Should this memory be evolved? Consider its relationships with other memories.
2. What specific actions should be taken (strengthen, update_neighbor)?
2.1 If choose to strengthen the connection, you must explicitly specify the neighbor memory IDs that are related to the new memory, and modify the new memory's content and keywords based on the content of those related neighbor memories.
2.2 If choose to update_neighbor, you can update the content and keywords of these memories based on the understanding of these memories. If the content and keywords are not updated, the new content and keywords be the same as the original ones. Generate the new content and keywords in the sequential order of the input neighbors.
Keywords should be determined by the content of these characteristic of these memories, which can be used to retrieve them later and categorize them.
Note that the length of new_content_neighborhood must exactly equal the number of input **The nearest neighbors memories**, and the length of new_keywords_neighborhood must must exactly equal the number of input **The nearest neighbors memories**. Each element in new_content_neighborhood and new_keywords_neighborhood must correspond one-to-one and in order with The nearest neighbors memories.

=== NEW MEMORIES TO PROCESS ===
{batch_memories}
Return your decisions as a JSON object with the following structure:
```
{{
   "decisions": [
      {{
         "memory_index": 0,
         "should_evolve": true/false,
         "actions": ["strengthen", "update_neighbor"],
         "suggested_connections": ["neighbor_memory_ids_1",...,"neighbor_memory_ids_n"],
         "content_to_update": "new_content",
         "keywords_to_update": ["keyword_1",..."keyword_n"],
         "new_content_neighborhood": ["new_content",...,"new_content"],
         "new_keywords_neighborhood": [["keyword_1",...,"keyword_n"],...["keyword_1",...,"keyword_n"]],
      }},
      ...
   ]
}}
```
IMPORTANT: Process ALL {num_memories} memories and return exactly {num_memories} decisions.

Table 10: Prompts of Schema Linking and Schema Updating.