Where to Look and What to Use: Retrieve–Localize–Generate for Long-Term Conversational Memory Question Answering

Yifan Wang[1,5]*, Xinkui Lin[2,3]*, Yongxiu Xu[2,3]†, Shen Gao[1,5]†, Ruochen Yang[2,3]
Kun Huang[4], Yubin Wang[2,3], Jie Wu[4], Wei Liu[4], Jian Luan[4], Hongbo Xu[2,3], Shuo Shang[1]†
[1]University of Electronic Science and Technology of China, Chengdu, China
[2]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[3]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[4]MiLM Plus, Xiaomi Inc., China
[5]State Key Laboratory of Internet Architecture, Tsinghua University, Beijing, China
{yifanwang993w,jedi.shang}@gmail.com, {linxinkui,xuyongxiu}@iie.ac.cn

Abstract

Retrieval-augmented generation (RAG) enables large language models (LLMs) to answer questions by accessing external knowledge and has been widely adopted for long-term conversational memory question answering. However, existing methods suffer from two key challenges: (1) fragmented evidence scattered across temporally distant sessions, and (2) noisy content within retrieved sessions that triggers the *lost-in-the-middle* effect. To address these challenges, we propose MemLoc, a unified **Retrieve–Localize–Generate** framework for long-term conversational memory QA. For retrieval, MemLoc decomposes each session into multi-granularity memory units and performs query routing via an inner-memory graph with entropy-based granularity selection. It further models cross-session semantic and temporal dependencies through a cross-memory graph, enabling coarse-to-fine retrieval of top-K relevant memory candidates. For localization, we introduce a reasoning-based evidence locator trained with Self-reflective Hint Policy Optimization (SHPO), which performs progressive refinement by extracting query-relevant fragments within memory units to suppress noise and reranking across candidates to remove redundancy, producing a compact evidence set with lightweight location IDs. For generation, these IDs act as precise grounding signals that guide the LLM to the correct memory positions, mitigating the lost-in-the-middle effect while preserving original contextual integrity. Extensive experiments on four benchmarks demonstrate that MemLoc achieves state-of-the-art retrieval accuracy and response quality while maintaining efficiency. Our code is available at: https://github.com/Nikol-coder/MemLoc.

*These authors contributed equally to this work. This work was done during an internship at Xiaomi Inc.

†Corresponding authors: Yongxiu Xu, Shen Gao and Shuo Shang.

1 Introduction

Recent advances in large language models (LLMs) (Li et al., 2026; Wang et al., 2024, 2025a) have enabled personalized conversational assistants for applications such as customer service and intelligent agents (Zhang et al., 2024; Du et al., 2025a). However, the limited context window of LLMs cannot accommodate continuously growing dialogue histories (Hsieh et al., 2024; Liu et al., 2024), making it difficult to maintain response consistency and personalization over time. To address this, retrieval-augmented memory systems have been widely adopted, where historical dialogues are stored as external memory and selectively retrieved to ground LLM responses (Qian et al., 2025; Yu et al., 2025a). Despite their promise, existing methods face two fundamental challenges when applied to long-term conversational question answering.

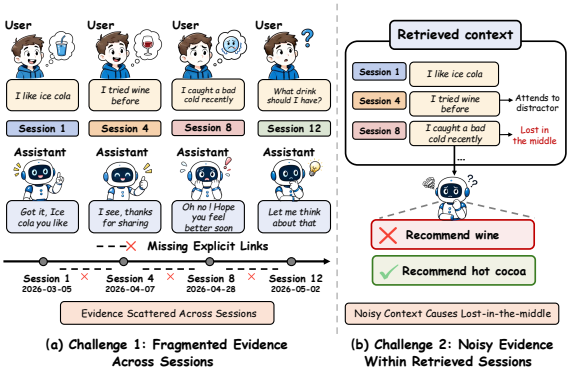

Figure 1: Challenges of Long-Term Memory Retrieval and Generation.

Fragmented Evidence Across Sessions. Relevant information is often scattered across multiple temporally distant sessions, yet most retrieval methods treat each session as an independent unit. Single-granularity approaches (Lu et al., 2023) fail to capture fine-grained cross-session associations, while multi-granularity methods (Rezazadeh et al.,

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 27680–27701
October 24-29, 2026 ©2026 Association for Computational Linguistics

2025; Xu et al., 2025a) still lack explicit structures to model the semantic and temporal relations between related events across sessions, leading to incomplete evidence aggregation for complex multi-hop queries.

Noisy Evidence Within Retrieved Sessions. Even when relevant sessions are successfully retrieved, they typically contain substantial irrelevant content alongside critical evidence. This triggers the *"lost-in-the-middle"* effect (Liu et al., 2024), where LLMs attend to partial or incorrect segments. Prior work attempts to mitigate this through filtering or memory compression (Xu et al., 2025a; Chen et al., 2025), but such operations inevitably introduce information loss or semantic drift, compromising the fidelity of the original dialogues.

To address these challenges, we propose MemLoc, a unified *Retrieve–Localize–Generate* framework for long-term conversational memory QA. **For retrieval**, MemLoc organizes each session into multi-granularity memory units and builds an inner-memory graph to adaptively route queries to the most discriminative granularity via entropy-based selection. It further constructs a cross-memory graph that captures semantic and temporal dependencies across sessions, enabling coarse-to-fine retrieval of top-K candidate memories. **For localization**, we introduce a reasoning-based evidence locator trained via Self-reflective Hint Policy Optimization (SHPO). The locator performs two-stage evidence localization: first extracting query-relevant segments within each memory unit (inner-memory extraction) to suppress noise, then reranking across candidate units (cross-memory reranking) to eliminate redundancy, yielding a compact refined memory subset and the corresponding evidence IDs. **For generation**, the evidence IDs serve as lightweight cues that anchor the generator to precise memory locations, mitigating the *lost-in-the-middle* effect while preserving the integrity of the original memory context.

Our contributions are summarized as follows:

- We propose MemLoc, a unified *Retrieve–Localize–Generate* framework for long-term conversational memory QA. It combines inner-memory routing with cross-memory propagation for coarse-to-fine retrieval, inner-memory evidence extraction with cross-memory reranking for noise-resistant evidence selection, and ID-based cues for faithful generation.
- We design a reasoning-based evidence locator trained via Self-reflective Hint Policy Optimization (SHPO), which contrasts the model's own correct and incorrect reasoning trajectories to self-distill strategy hints without answer leakage, enabling progressive and noise-resistant evidence localization.
- Extensive experiments on four long-term memory QA benchmarks demonstrate that MemLoc achieves state-of-the-art retrieval accuracy and response quality while maintaining computational efficiency across different generator backbones.

2 Related Work

Long-Term Memory Retrieval and Generation. To enable personalized conversational agents, large language models (LLMs) retrieve relevant historical dialogues as external memory to ground response generation. Existing approaches mainly differ in how memory is represented and organized. Single-granularity methods treat each session as an atomic retrieval unit (Zhong et al., 2024), while multi-granularity approaches construct hierarchical structures to better capture contextual dependencies within sessions (Xu et al., 2025a; Lu et al., 2023). Despite these advances, long-term conversational memory QA faces two fundamental limitations. First, relevant evidence is often distributed across temporally distant sessions, yet existing methods typically lack explicit mechanisms to model cross-session relationships, leading to incomplete evidence aggregation for complex queries. Second, retrieved memories often contain substantial redundancy and irrelevant content, which can interfere with downstream reasoning and trigger the *lost-in-the-middle* effect. Prior efforts on memory compression or refinement (Chen et al., 2025) attempt to mitigate this issue, but they often risk discarding fine-grained details or introducing semantic drift. These limitations motivate more structured memory modeling and finer-grained evidence selection for long-term conversational reasoning.

3 Method

3.1 Overview

In this section, we formalize the task of long-term conversational memory question answering and introduce our MemLoc framework (Fig. 2). Our framework consists of two core components: (1) **M**emory Retrieval with **M**ulti-granularity

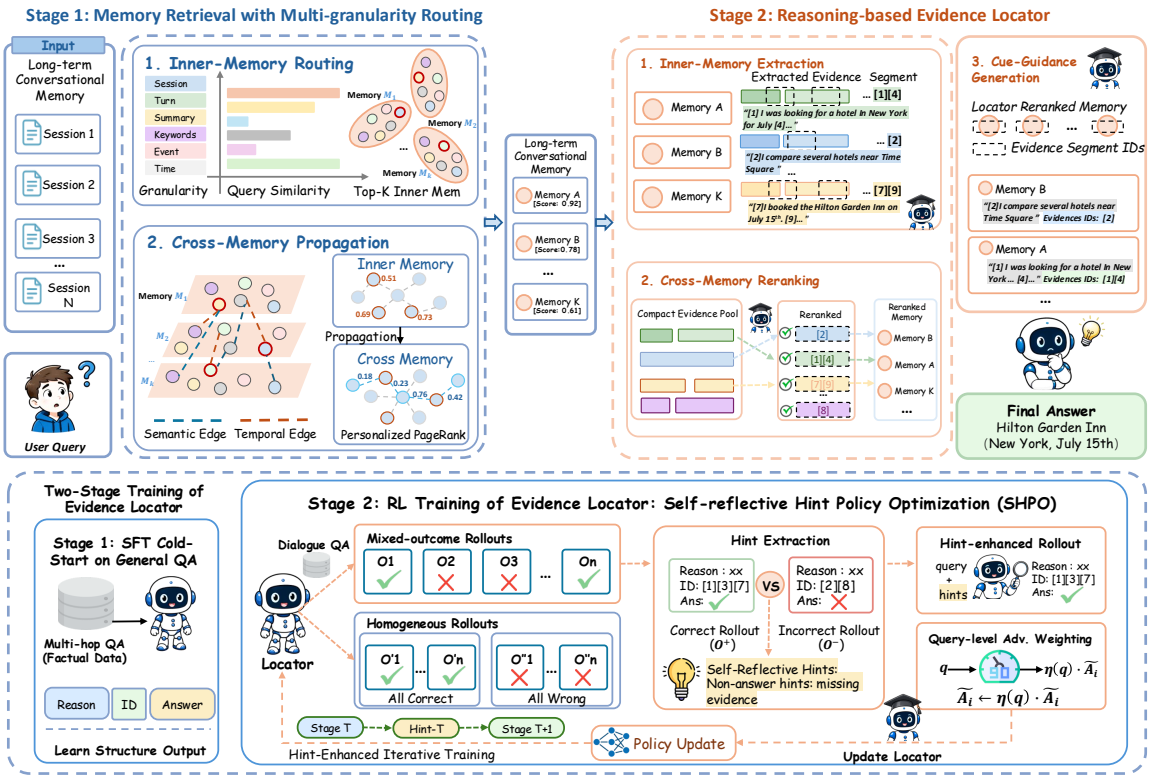

Figure 2: The overall framework of our MemLoc.

Routing (**MMR**, §3.3), and (2) **R**easoning-based **E**vidence **L**ocator (**REL**, §3.4).

3.2 Task Formulation

We formalize the problem as **Memory-Based Question Answering** following a *Retrieve–Localize–Generate* pipeline. Given a query q and a corpus of historical memory units $\mathcal{M} = \{M_1, M_2, \ldots, M_N\}$, the objective is to (1) retrieve the most relevant memory units, (2) localize critical evidence within the retrieved units, and (3) generate a faithful answer grounded in the localized evidence.

Each memory unit M_i contains a temporally coherent span of multi-turn interactions, stored as a sequence of dialogue segments:

$$M_i = \{u_{i,1}, u_{i,2}, \ldots, u_{i,T_i}\}.$$

The task decomposes into three stages:
(1) **Retrieve.** A retrieval function $\mathcal{R}$ selects the top-K memory units most relevant to the query q from the full corpus $\mathcal{M}$:

$$\mathcal{M}_K(q) = \mathcal{R}(q, \mathcal{M}), \quad |\mathcal{M}_K(q)| = K. \tag{1}$$

(2) **Localize.** An evidence locator $\mathcal{L}$ pinpoints the critical evidence segments within the retrieved units. It first suppresses irrelevant content within each unit (inner-memory extraction), then eliminates redundancy across units (cross-memory reranking), yielding a refined subset $\mathcal{M}_{K'}(q) \subseteq \mathcal{M}_K(q)$ with $K' \leq K$ and a set of evidence IDs that index the supporting segments:

$$\big(\mathcal{M}_{K'}(q),\ \mathcal{I}\big) = \mathcal{L}\big(q,\ \mathcal{M}_K(q)\big), \tag{2}$$

where $\mathcal{I} = \{eids_i \mid M_i \in \mathcal{M}_{K'}(q)\}$ collects the local evidence IDs for each retained memory unit.
(3) **Generate.** A generative model f_θ produces the final answer a by reasoning over the query q, the refined memory units $\mathcal{M}_{K'}(q)$, and the evidence IDs $\mathcal{I}$:

$$a = f_\theta\big(q,\ \mathcal{M}_{K'}(q),\ \mathcal{I}\big). \tag{3}$$

The evidence IDs serve as lightweight, interpretable cues that anchor the generator to precise memory locations.

3.3 Memory Retrieval with Multi-granularity Routing

We cast conversational memory retrieval as a two-stage process: (1) *Inner-Memory Routing*, which identifies the most query-relevant fine-grained memory unit within each session; and (2) *Cross-Memory Propagation*, which propagates relevance

signals along semantic and temporal connections across sessions, yielding a coarse top-K session candidate set.

3.3.1 Multi-Granularity Memory Representation

Each session s_i is decomposed into six complementary levels of granularity:

$$s_i = \left(m_i^{\text{session}}, m_i^{\text{turn}}, m_i^{\text{summary}}, m_i^{\text{keyword}}, m_i^{\text{event}}, m_i^{\text{time}}\right), \tag{4}$$

representing session-level, turn-level, summary, keyword, event, and temporal information. Each granularity is encoded into a dense vector $\mathbf{e}_i^g \in \mathbb{R}^d$ using the embedding model, providing a unified multi-view representation of the session.

3.3.2 Stage 1: Inner-Memory Routing

Queries may target factual details, temporal cues, or broad topics, each naturally aligning with different memory granularities. To adaptively select the most informative granularity per query, we introduce an entropy-based routing mechanism (Thiombiano et al., 2025).

Given a query embedding $\mathbf{q}$, we compute the similarity between $\mathbf{q}$ and each memory unit of granularity g:

$$s_i^g = \mathbf{q} \cdot (\mathbf{e}_i^g)^\top. \tag{5}$$

We apply a temperature-scaled softmax over all sessions to obtain a relevance distribution $p^g = \text{softmax}_i(s^g/\tau)$ The sharpness of this distribution is quantified by Shannon entropy:

$$H^g = -\sum\nolimits_j p_j^g \log p_j^g. \tag{6}$$

A low H^g indicates that only a few sessions receive high probability, i.e., granularity g provides a peaked and discriminative signal. We normalize the entropy by the logarithm of the number of sessions to account for different candidate-set sizes, and define the routing weight as

$$w^g = \text{softmax}_g\left(1 - \frac{H^g}{\log N_g}\right), \tag{7}$$

where N_g denotes the number of memory units at granularity g. This formulation assigns larger weights to granularities with lower normalized entropy while guaranteeing non-negative weights. The inner-memory relevance of session i is then computed as the weighted sum of its unit-level similarities:

$$r_i = \sum_{g=1}^{6} w^g \cdot s_i^g. \tag{8}$$

This can be interpreted as a soft selection over memory units within each session, where the resulting score r_i serves as a seed signal for cross-session propagation in the next stage.

3.3.3 Stage 2: Cross-Memory Propagation

Conversational sessions are inherently interdependent, as they share recurring entities, semantic themes, and temporal continuity. This implies that a memory unit highly relevant to a query often serves as a strong indicator of related evidence distributed across other sessions connected via semantic or temporal relations. To capture these dependencies, we construct a memory graph over memory units and propagate relevance signals using Personalized PageRank (Yang et al., 2024), enabling global information diffusion across sessions.

3.3.4 Memory Graph Construction

We build an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each node is a single memory unit m_i^g. Edges encode two types of cross-session relations:

Semantic edges. We link memory units of the same granularity that exhibit high semantic similarity, using an adaptive threshold based on Gaussian Mixture Models (GMM) (Rasmussen, 1999). For each node, we compute its cosine similarity against all existing same-granularity nodes, take the top-K_{cand} candidates, and fit a two-component GMM to their similarity scores. Only neighbors whose similarity belongs to the higher-mean component are retained.

Temporal edges. To capture chronological order, we connect the corresponding granularity units of consecutively indexed sessions:

$$\mathcal{E}_{\text{temp}} = \{\, (m_i^{(g)}, m_{i+1}^{(g)}) \mid \forall\, g \in \{1, \ldots, 6\},\ \forall\, i \,\}, \tag{9}$$

allowing information to flow along the timeline.

3.3.5 Propagation via Personalized PageRank

We construct an initial relevance vector $\mathbf{r}$ by taking the unit-level scores $r_i^g = w^g \cdot s_i^g$ (which compose r_i in Eq. 8) and retaining only the top-K_0 entries, setting the rest to zero. This sparsification preserves the strongest query-responsive units and reduces computation. We then propagate these signals over $\mathcal{G}$ using Personalized PageRank (PPR):

$$\boldsymbol{\pi} = (1 - d) \cdot \mathbf{A}\,\boldsymbol{\pi} + d \cdot \mathbf{r}, \tag{10}$$

where $\mathbf{A}$ is the column-stochastic transition matrix defined as

$$\mathbf{A}_{ij} = \begin{cases} \frac{1}{|\mathcal{N}(j)|}, & \text{if } (i,j) \in \mathcal{E}, \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

$\mathcal{N}(j)$ is the neighbor set of node j, and d is the restart probability. The stationary distribution $\boldsymbol{\pi}$ assigns a final relevance score π_i^g to each memory unit, effectively diffusing the query's attention along both semantic associations and temporal flows. Finally, session-level scores are obtained by aggregating node scores:

$$S_i = \sum_{g=1}^{6} \pi_i^g. \tag{12}$$

The top-K sessions ranked by S_i form the coarse recall set, which is subsequently passed to the fine-grained localization and answer generation module.

3.4 Reasoning-based Evidence Locator

Recent methods (Li et al., 2025; Nan et al., 2025) employ LLM-based filters to reduce redundancy by selecting query-relevant segments from individual memory units. However, in long-term conversational memory QA, critical evidence is often distributed across temporally dispersed dialogue sessions and requires aggregation to form complete answers. Existing filters, primarily designed for single-session contexts, lack mechanisms to identify and synthesize cross-session evidence, frequently missing crucial contextual links.

To address this limitation, we propose a reasoning-based evidence Locator that operates over retrieved memory units via a two-level design: Inner-Memory Extraction and Cross-Memory Reranking. At the inner-memory level, the Locator is applied to the candidate memory units retrieved by MMR to remove irrelevant content within each unit. At the cross-memory level, the Locator leverages the compact evidence to rerank candidate memory units. This evidence-guided reranking yields a final subset of memory units, which is used as input to the generator.

Inner-Memory Extraction. To yield a structured inner-memory representation, each retrieved memory unit $M_i \in \mathcal{M}_K(q)$ is decomposed into fine-grained atomic evidence segments, each with a unique local evidence ID:

$$\mathbf{M}_i = \{m_{i,1}, m_{i,2}, \ldots, m_{i,L_i}\}$$

where $m_{i,l}$ denotes the l-th atomic evidence segment extracted from memory M_i, and L_i is the total number of atomic segments within memory unit M_i.

For each retrieved memory unit $\mathbf{m}_i$, the Locator identifies relevant evidence by outputting a set of *local evidence IDs*:

$$eid_{s_i} = \text{Locator}(q, \mathbf{M}_i), \quad eid_{s_i} \subseteq \{1, \ldots, L_i\}, \tag{13}$$

where each ID corresponds to atomic evidence segments directly relevant to the query. Then we construct a purified evidence block by concatenating only the selected evidence units:

$$B_i = \{\, m_{i,t} \mid t \in eid_{s_i} \,\}, \tag{14}$$

This inner-memory evidence extraction operates on all K retrieved memories, effectively eliminating irrelevant content within each memory unit.

Cross-Memory Reranking. We first renumber and integrate all evidence blocks B_i into a compact cross-memory representation:

$$M^{\text{in}} = \{B_1, B_2, \ldots, B_K\}, \tag{15}$$

The Locator then processes M^{in} to identify the minimal evidence chain required for answering q:

$$sid_s = \text{Locator}(q, M^{\text{in}}), \quad sid_s \subseteq \{1, \ldots, K\}, \tag{16}$$

$$\mathcal{M}_{K'}(q) = \{\, M_i \mid i \in sid_s \,\}, \tag{17}$$

which retains the essential evidence chains while eliminating redundant information. Finally, we use $\mathcal{M}_{K'}(q)$ to rerank the original top-K candidate memory units, producing reranked memory sets.

Cue-Guidance Generation. Following cross-memory reranking, we provide the generator with both the selected memory units $\mathcal{M}_{K'}(q)$ and their corresponding evidence IDs:

$$eid_s = \{\, eid_{s_i} \mid M_i \in \mathcal{M}_{K'}(q) \,\}. \tag{18}$$

The final response is then generated as:

$$a = \text{Generator}\big(q, \mathcal{M}_{K'}(q), eid_s\big). \tag{19}$$

These evidence IDs serve as lightweight, interpretable cues that directly anchor the generator to critical evidence segments within memory.

3.5 Two-Stage Training of Evidence Locator

3.5.1 Stage 1: SFT Cold-Start

To bootstrap the Locator with structured reasoning and accurate evidence localization before RL, we first perform supervised fine-tuning (SFT) on multi-hop QA datasets, including HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), and 2WikiMultihopQA (Ho et al., 2020). We synthesize structured reasoning demonstrations: given a query q, candidate evidence set $\mathcal{M} = \{m_1, \ldots, m_k\}$, and gold answer a, the oracle produces a target tuple $y = \langle r, \mathcal{I}, a \rangle$, where r is a chain-of-thought trace explicitly referencing evidence content, $\mathcal{I} \subseteq \{1, \ldots, k\}$ is the set of supporting evidence indices, and a is the answer. We then fine-tune the Locator on $\mathcal{D}_{\text{sft}} = \{(x, y)\}$ with $x = (q, \mathcal{M})$, minimizing:

$$\mathcal{L}_{\text{sft}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{sft}}} \left[\log p_\theta(y \mid x)\right]. \tag{20}$$

This stage instills structured reasoning and evidence localization capabilities, providing a stable starting point for the RL phase.

3.5.2 Stage 2: Self-reflective Hint Policy Optimization

While SFT equips the Locator with basic reasoning and evidence localization capabilities, long-term dialogue QA poses additional challenges that require targeted adaptation: (1) redundant evidence, (2) conflicting memory units, and (3) multi-turn dependencies. To address these, we propose **Self-reflective Hint Policy Optimization (SHPO)**, a novel reinforcement learning strategy that lets the model act as its own teacher by contrasting its successful and failed trajectories on the same sample, and then injects the resulting hints into the next training iteration. We conduct SHPO on out-of-domain dialogue datasets such as TimeDialogue (Du et al., 2025b; Qin et al., 2021) and HaluMem (Chen et al., 2026), which provide diverse, long-term conversations for learning robust hint generation and evidence refinement.

Reward Design. We combine three complementary reward signals to jointly optimize structured reasoning, evidence localization, and answer correctness.

- **Format Reward** r_{fmt}: equals 1 if the output contains correctly delimited tags (`<reason>`, `<id>`, `<answer>`), and 0 otherwise.
- **Evidence Reward** r_{evid}: measures the accuracy of evidence localization by comparing the predicted evidence index set $\mathcal{I}_{\text{pred}}$ against the gold set $\mathcal{I}_{\text{gold}}$:

$$r_{\text{evid}} = \frac{|\mathcal{I}_{\text{pred}} \cap \mathcal{I}_{\text{gold}}|}{|\mathcal{I}_{\text{gold}}|}. \tag{21}$$

- **Answer Reward** r_{ans}: equals 1 if the gold answer a_{gold} is a substring of the predicted answer a_{pred}, and 0 otherwise.

$$r_{\text{ans}} = \mathbb{I}\big(a_{\text{gold}} \subseteq a_{\text{pred}}\big), \tag{22}$$

The final reward is computed as a weighted combination:

$$r = \alpha \cdot r_{\text{ans}} + \beta \cdot r_{\text{evid}} + (1 - \alpha - \beta) \cdot r_{\text{fmt}}, \tag{23}$$

where α and β balance answer correctness and evidence localization quality.

Mixed Rollout and Hint Extraction. Given a query q and memory set $\mathcal{M}$, we sample G rollouts $\{o_i\}_{i=1}^{G}$ from the current policy π_θ; each outputs a structured tuple $\langle r, \mathcal{I}, a \rangle$. A rollout is *correct* if $r_{\text{evid}} = 1$ (Eq. 21), and *incorrect* otherwise. We focus on **mixed-outcome** samples: those where at least one rollout is correct and at least one is incorrect. For each such instance, the policy itself acts as a teacher $\mathcal{T}$: it contrasts correct trajectories o^+ with incorrect ones o^- and distills a concise *non-answer hint* h that identifies missing or misused evidence without revealing the gold answer:

$$h = \mathcal{T}(o^+, o^-). \tag{24}$$

Since both trajectories come from the same model, the hint captures decision-level gaps that are immediately actionable for the current policy.

Hint-Enhanced Iterative Training. Let $x^{(0)}$ denote the original prompt specifying the structured generation task. At training stage t, for each mixed-outcome query we construct an augmented prompt by appending the distilled hint $h^{(t)}$:

$$x^{(t+1)} = x^{(t)} \oplus h^{(t)}. \tag{25}$$

Non-mixed samples retain their prompts unchanged, as they offer limited contrastive signal. We then perform GRPO on the updated prompt set; the injected hints steer the policy toward mislocalized evidence patterns exposed by rollout discrepancies. Iterating this procedure yields a self-improving curriculum where the policy progressively sharpens its evidence localization by learning from its own success-failure contrasts.

Table 1: Best results are shown in **bold**, second-best are underlined. **4o-J**: GPT-4o-as-Judge; **F1**: token-level F1; **Tokens**: average tokens per query. All methods use GPT-4o mini as the generator.

Model	LongMemEval-S			LongMemEval-M			LoCoMo			Long-MT-Bench+		
	4o-J	F1	Tokens	4o-J	F1	Tokens	4o-J	F1	Tokens	4o-J	F1	Tokens
Full History	50.60	11.48	103k	12.20	5.70	128k	33.43	12.23	20.1k	67.44	36.07	19.2k
RAPTOR	32.20	12.08	6,254		*Timeout*		31.72	14.55	1,931	59.72	37.69	10.6k
SeCom	56.00	12.95	2,741	42.80	11.33	2,821	44.21	13.79	1,021	64.58	36.68	4,714
A-Mem	55.60	13.73	9,018		*Timeout*		40.81	14.72	3,042	65.73	36.82	13.7k
HippoRAG 2	57.60	14.73	8,530		*Timeout*		45.62	16.66	2,991	63.54	35.64	13.6k
Mem0	42.00	17.72	6,787	32.00	14.38	7,452	36.39	8.16	2,450	38.33	25.42	8,100
MemGAS	60.20	20.38	8,829	45.40	16.85	8,852	41.07	17.66	2,825	69.44	41.49	12.9k
MemoryAgent	42.80	11.71	11.2k	34.20	9.58	11.8k	38.72	14.80	6,800	39.03	11.69	16.2k
Memory-R1	45.60	21.49	13.5k	29.00	5.95	14.2k	38.07	5.52	7,900	55.90	8.46	18.4k
RecurSum	35.40	12.29	8,853	23.80	10.04	8,927	22.56	9.14	3,074	24.65	26.58	13.5k
MemoryTree	41.40	12.02	7,600	35.80	11.57	8,200	35.21	8.95	5,300	51.39	30.13	11.5k
MemSkill	59.14	15.61	6,900	42.20	14.88	7,300	47.21	17.37	4,900	62.33	32.01	10.7k
MemLoc (4B)	65.80	16.80	8,645	48.20	13.41	8,326	57.20	18.13	2,744	73.96	37.98	12.5k
MemLoc (8B)	**68.40**	**21.52**	8,941	**54.20**	**17.27**	8,537	**57.40**	**19.99**	2,907	**77.42**	**46.65**	13.9k

Policy Optimization. We optimize π_θ via GRPO, re-weighting advantages toward mixed-outcome queries. For a group of G rollouts $\{o_i\}$ from query q, the standard group-normalized advantage $\hat{A}_{i,t}$ is scaled by a query-dependent factor:

$$\tilde{A}_{i,t} = \eta(q) \cdot \hat{A}_{i,t}, \tag{26}$$

with $\eta(q) = 1$ for mixed-outcome queries, and a smaller value for homogeneous ones. A KL penalty to the SFT reference π_{sft} prevents forgetting. The per-stage objective is

$$\mathcal{J}^{(t)}(\theta) = \mathbb{E}_{q,\mathcal{M}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\min\big(\rho_i\tilde{A}_i,\ \text{clip}(\rho_i, 1-\epsilon, 1+\epsilon)\tilde{A}_i\big) - \gamma D_{\text{KL}}(\pi_\theta \| \pi_{\text{sft}})\Bigg], \tag{27}$$

where $\rho_i = \frac{\pi_\theta(o_i|q,\mathcal{M})}{\pi_{\text{old}}(o_i|q,\mathcal{M})}$ is the importance sampling ratio and ϵ is the clipping parameter.

4 Experiment

4.1 Datasets and Evaluation Metrics

We evaluate our method on four long-term memory QA benchmarks: *LongMemEval-S (LME-S)*, *LongMemEval-M (LME-M)* (Wu et al., 2025), *LoCoMo* (Maharana et al., 2024), and *Long-MT-Bench+* (Pan et al., 2025). These benchmarks assess the long-term conversational memory capabilities of LLM agents. We evaluate both **retrieval** and **generation** performance. For retrieval, we report **Recall@K** and **NDCG@K** to measure retrieval coverage and ranking quality, respectively. For generation, we primarily adopt LLM-as-a-Judge evaluation using **GPT-4o** (Zheng et al., 2023a) as the main evaluation protocol, assessing factual consistency, semantic alignment, and response quality with respect to reference answers. In addition, we report conventional automatic metrics, including **F1**, **BLEU**, and **BERTScore**, as complementary indicators (Xu et al., 2025a; Maharana et al., 2024; Pan et al., 2025). More comprehensive experimental results and analyses are provided in the Table 13.

4.2 Baselines

We compare MemLoc against four categories of baselines. **(0) Full-context reference.** This method answers each query by processing the entire conversation history in chronological order. **(1) Structured RAG models** construct external knowledge structures (graphs or trees) from raw conversations and retrieve over them: RAPTOR (Sarthi et al., 2024), HippoRAG 2 (Gutiérrez et al., 2025), and MemoryTree (Rezazadeh et al., 2025). **(2) Memory-based methods** explicitly maintain a long-term memory store using summarization, semantic linking, topic segmentation, or multi-granularity indexing, and retrieve relevant entries at query time: RecurSum (Wang et al., 2025b), A-Mem (Xu et al., 2025b), SeCom (Pan et al., 2025), MemGAS (Xu et al., 2025a), and Mem0 (Chhikara et al., 2025). **(3) RL-based memory methods** train an LLM agent with reinforcement learning to actively manage or compress memory: Memory-R1 (Yan et al., 2026), MemoryAgent (Yu et al., 2025b), and MemSkill (Zhang et al., 2026).

4.3 Main Results

Question Answering Performance. The long-term memory QA results are shown in Table 1. We observe that: (1) MemLoc achieves state-of-the-art question answering performance on the 4o-J and

F1 metrics across all benchmarks, consistently outperforming existing methods while maintaining effective token consumption. Notably, both variants of our locator demonstrate strong and robust performance across datasets, with the 4B model already achieving competitive results and the 8B model further delivering state-of-the-art performance, highlighting the scalability and effectiveness of our method. (2) Although RL-based methods exhibit strong performance on general QA tasks, memory-based methods consistently outperform RL-based memory-management methods in long-term conversational memory QA, suggesting that explicit memory mechanisms are better suited for capturing and retrieving information across extended conversational histories. Further analysis is provided in the Appendix.

Table 2: Retrieval performance. Full combines MMR and REL; "w/o" removes the specified component. R@k: Recall@k; N@k: NDCG@k.

Method	R@3	N@3	R@5	N@5	R@10	N@10
LongMemEval-S						
SeCom	71.06	80.88	80.43	83.08	89.15	85.11
MemGAS	78.51	**86.83**	88.94	**88.77**	94.47	**89.96**
MemLoc (MMR)	83.90	81.02	90.35	83.38	94.50	84.02
MemLoc (Full)	**87.97**	86.81	**92.56**	88.62	**94.50**	89.44
w/o IM-Routing	86.02	83.36	89.98	85.09	92.60	87.72
w/o CM-Prop.	85.27	83.65	89.23	84.92	92.95	85.55
only SFT	86.45	82.40	92.53	84.77	94.50	85.57
only RL	87.25	86.37	92.08	88.08	94.50	89.03
LoCoMo						
SeCom	55.24	57.90	64.80	62.36	78.30	66.97
MemGAS	57.30	58.76	67.32	63.62	81.82	68.42
MemLoc (MMR)	66.84	54.58	77.10	58.75	90.09	63.83
MemLoc (Full)	**77.85**	**69.76**	**81.72**	**71.67**	**90.09**	**73.80**
w/o IM-Routing	61.61	53.15	72.87	57.88	81.10	62.88
w/o CM-Prop.	63.57	56.00	75.20	60.81	85.70	65.18
only SFT	77.16	69.66	81.58	71.40	90.09	73.58
only RL	64.88	58.74	75.10	62.97	90.09	67.23

Retrieval Performance. Table 2 reports retrieval results on LongMemEval-S and LoCoMo. MemLoc surpasses SeCom and MemGAS, demonstrating the effectiveness of its multi-granularity design that combines inner-memory routing with cross-memory propagation. Removing either inner-memory routing (w/o IM-Routing) or cross-memory propagation (w/o CM-Prop.) consistently degrades recall, confirming that both components are indispensable for locating scattered evidence across sessions. When augmented with REL as an evidence-aware reranker, MemLoc achieves the best scores, showing that fine-grained evidence selection further refines the retrieval ranking. Within REL, using only SFT or only RL underperforms the full two-stage training, demonstrating that both supervised cold-start and self-reflective hint optimization are critical for building a reliable locator.

4.4 Ablation Study

4.4.1 Evidence Localization

Table 3: Ablation study of evidence localization, with GPT-4o-as-Judge (4o-J) as the evaluation metric.

Method	LME-M	LME-S	LoCoMo
MemLoc (Evidence only)	36.40	56.60	41.33
MemLoc w/ MMR	44.20	65.60	45.87
+ Extract	45.60	66.00	47.13
+ Extract + Rerank	48.40	66.60	57.00
+ Extract + Rerank + Cue	**51.80**	**67.20**	**57.40**

We evaluate the contribution of each component in our evidence localization module, including *Extract*, *Rerank*, and *Cue*. Using only extracted evidence as model input performs poorly, indicating that discarding contextual memory leads to significant information loss. Starting from MMR, we progressively add each component. *Extract* improves performance by filtering irrelevant content within retrieved memory units. *Rerank* further improves performance by refining cross-memory relevance estimation based on compact evidence. Finally, *Cue* provides explicit guidance to the generator, leading to the best performance across all benchmarks. Overall, each component contributes complementary benefits, and their combination yields the most effective evidence localization and utilization.

4.4.2 Self-reflective Hint Policy Optimization

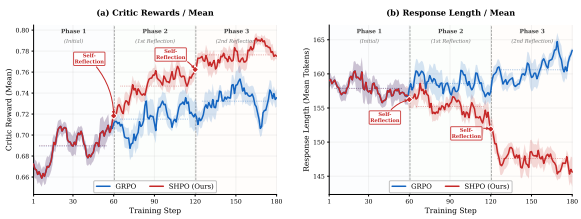

Figure 3: Training dynamics comparison between GRPO and SHPO on the 8B locator.

As shown in Fig. 3, SHPO introduces self-reflective hints on mixed-outcome samples at step 60, leading to a noticeable improvement in mean reward during subsequent training. Meanwhile, the average response length decreases significantly, indicating fewer redundant descriptions of irrelevant memory fragments and more precise evidence localization.

Table 4: Training ablation of the locator on LongMemEval-S. R@k: Recall@k; N@k: NDCG@k.

Method	R@3	N@3	R@5	N@5	R@10	N@10
No locator	83.90	81.02	90.35	83.38	94.50	84.02
Base (untrained)	84.62	81.91	91.68	84.18	94.50	84.91
Only SFT	86.45	82.40	92.53	84.77	94.50	85.57
Only RL (GRPO)	86.98	84.71	92.03	86.42	94.50	86.68
Only RL (SHPO)	87.25	86.37	92.08	88.08	94.50	89.03
Full (SFT+GRPO)	87.63	85.46	92.47	87.34	94.50	88.12
Full (SFT+SHPO)	**87.97**	**86.81**	**92.56**	**88.62**	94.50	**89.44**

As shown in Table 4, even without training, the locator improves R@3 from 83.90 to 84.62 over the no-locator baseline, demonstrating that the gain primarily comes from the retrieve–localize–generate framework rather than additional QA supervision. Training further improves evidence localization, with SHPO achieving the best overall performance.

4.5 Statistical Significance

To assess the stability of our results, we conduct three independent runs with different random seeds. We report the mean, standard deviation, and 95% confidence intervals on both LongMemEval-S and LoCoMo. As shown in Table 5, the small standard deviations demonstrate stable performance across runs, while the confidence intervals remain well above those of other methods, further confirming the robustness of our improvements.

Table 5: Mean $\pm$ SD and 95% CI over three runs.

Setting	4o-J $\pm$ SD	95% CI	F1 $\pm$ SD
LongMemEval-S	68.40 ± 0.62	[66.86, 69.94]	21.52 ± 0.35
LoCoMo	57.40 ± 0.42	[56.36, 58.44]	19.99 ± 0.25

5 Conclusion

In this paper, we propose MemLoc, a Retrieve–Localize–Generate framework for long-term conversational memory QA, where both retrieval and localization are built upon inner-memory and cross-memory mechanisms. For retrieval, we first organize each session into multi-granularity memory units and construct an inner-memory graph to route queries to the most relevant memory granularity. We then model semantic and temporal relations across sessions via a cross-memory graph, enabling coarse-to-fine retrieval of top-K candidate memories from long dialogue histories. For localization, we train an evidence locator via self-reflective hint optimization. The locator first extracts query-relevant evidence within each memory unit to suppress noise and irrelevant content, and then performs cross-memory reranking to remove redundant evidence across sessions. The resulting evidence is used to guide response generation with precise memory grounding. For generation, the generator is guided by lightweight ID-based cues that anchor it to the critical evidence within retrieved memories, yielding faithful answers. Extensive experiments on multiple benchmarks demonstrate the effectiveness and robustness of our approach.

Limitations

Our framework introduces additional computational overhead from cross-session graph construction, propagation, and coarse-to-fine retrieval, which may increase latency for extremely long conversations. Although top-K sparsification and adaptive GMM thresholding mitigate this cost, retrieval remains slower than flat vector search. Given the substantial improvements in multi-hop evidence localization and noise resilience, this latency trade-off is justified. Notably, our framework is architecture-agnostic and can be seamlessly integrated with stronger base LLMs or embedders, with performance expected to scale as foundational capabilities advance. Future research will focus on incremental graph maintenance, online retrieval acceleration, and joint end-to-end optimization of the retriever and locator.

Ethics Statement

In this paper, our experiments use only publicly available long-term memory QA benchmarks and open-source dialogue datasets. All models and APIs are used in accordance with their respective licensing and usage terms, and no private or sensitive user conversation data are involved. Therefore, we affirm that our methodology and generated outputs adhere to responsible AI practices and do not compromise user privacy or data ownership.

Acknowledgement

This paper was supported by the National Key Research and Development Program of China (No.2025YFF0730600, No.2024YFF1401300), the National Natural Science Foundation of China (No.U25B2049, 62432002 and 62406061), and the State Key Laboratory of Internet Architecture, Tsinghua University (No. HLW2025MS10).

References

Ding Chen, Simin Niu, Kehang Li, Peng Liu, Xiangping Zheng, Bo Tang, Xinchi Li, Feiyu Xiong, and Zhiyu Li. 2026. Halumem: Evaluating hallucinations in memory systems of agents. *Preprint*, arXiv:2511.03506.

Guanzheng Chen, Xin Li, Michael Qizhe Shieh, and Lidong Bing. 2025. Longpo: Long context self-evolution of large language models through short-to-long preference optimization. *Preprint*, arXiv:2502.13922.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2023. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2309.07597.

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

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z. Pan. 2025a. Rethinking memory in llm based agents: Representations, operations, and emerging topics. *Preprint*, arXiv:2505.00675.

Yiming Du, Baojun Wang, Yifan Xiang, Zhaowei Wang, Wenyu Huang, Boyang Xue, Bin Liang, Xingshan Zeng, Fei Mi, Haoli Bai, Lifeng Shang, Jeff Z. Pan, Yuxin Jiang, and Kam-Fai Wong. 2025b. Memory-t1: Reinforcement learning for temporal reasoning in multi-session agents. *Preprint*, arXiv:2512.20092.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2025. Hipporag: Neurobiologically inspired long-term memory for large language models. *Preprint*, arXiv:2405.14831.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *Preprint*, arXiv:2404.06654.

Xiaopeng Li, Pengyue Jia, Derong Xu, Yi Wen, Yingyi Zhang, Wenlin Zhang, Wanyu Wang, Yichao Wang, Zhaocheng Du, Xiangyang Li, and 1 others. 2026. A survey of personalization: From rag to agent. *ACM Transactions on Information Systems*, 44(4):1–39.

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, and 20 others. 2025. Memos: A memory os for ai system. *Preprint*, arXiv:2507.03724.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *Preprint*, arXiv:2308.08239.

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.

Jiayan Nan, Wenquan Ma, Wenlong Wu, and Yize Chen. 2025. Nemori: Self-organizing agent memory inspired by cognitive science. *Preprint*, arXiv:2508.03341.

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

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. *Preprint*, arXiv:2409.05591.

Lianhui Qin, Aditya Gupta, Shyam Upadhyay, Luheng He, Yejin Choi, and Manaal Faruqui. 2021. TimeDial: Temporal Commonsense Reasoning in Dialog. In *Proc. of ACL*.

Carl Edward Rasmussen. 1999. The infinite gaussian mixture model. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'99, page 554–560, Cambridge, MA, USA. MIT Press.

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. 2025. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms. *Preprint*, arXiv:2410.14052.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. *Preprint*, arXiv:2401.18059.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Abdoul Majid O. Thiombiano, Brahim Hnich, Ali Ben Mrad, and Mohamed Wiem Mkaouer. 2025. Moxe: Mixture of xlstm experts with entropy-aware routing for efficient language modeling. *Preprint*, arXiv:2505.01459.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Bing Wang, Xinnian Liang, Jian Yang, Hui Huang, Shuangzhi Wu, Peihao Wu, Lu Lu, Zejun Ma, and Zhoujun Li. 2025a. Scm: Enhancing large language model with self-controlled memory framework. *Preprint*, arXiv:2304.13343.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024. A survey on large language model based autonomous agents. *Frontiers of computer science*, 18(6):186345.

Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. 2025b. Recursively summarizing enables long-term dialogue memory in large language models. *Preprint*, arXiv:2308.15022.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *Preprint*, arXiv:2410.10813.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, wenlin zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. 2025a. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. *Preprint*, arXiv:2505.19549.

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

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. 2026. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *Preprint*, arXiv:2508.19828.

Mingji Yang, Hanzhi Wang, Zhewei Wei, Sibo Wang, and Ji-Rong Wen. 2024. Efficient algorithms for personalized pagerank computation: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(9):4582–4602.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. 2025a. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *Preprint*, arXiv:2507.02259.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and 1 others. 2025b. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. 2026. Memskill: Learning and evolving memory skills for self-evolving agents. *Preprint*, arXiv:2602.02474.

Kai Zhang, Yangyang Kang, Fubang Zhao, and Xiaozhong Liu. 2024. LLM-based medical assistant personalization with short- and long-term memory coordination. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2386–2398, Mexico City, Mexico. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023a. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Yaowei Zheng, Junting Lu, Shenzhi Wang, Zhangchi Feng, Dongdong Kuang, and Yuwen Xiong. 2025. Easyr1: An efficient, scalable, multi-modality rl training framework. `https://github.com/hiyouga/EasyR1`.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 3: system demonstrations)*, pages 400–410.

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.

Algorithm 1 Self-reflective Hint Policy Optimization (SHPO)

Require: SFT policy $\pi_\theta^{(0)}$, training queries $\mathcal{Q}$, stages T, rollouts G, scaling factor η_{homo}
Ensure: Locator policy $\pi_\theta^{(T)}$

```
Initialize prompts: x_q^(0) ← x, ∀q ∈ Q
for t = 0, ..., T − 1 do
    for each query q with prompt x_q^(t) do
        Sample G rollouts {o_i} from π_θ^(t)(· | x_q^(t))
        Compute token-level rewards r_{i,t} and group-normalized advantages Â_{i,t}
        Determine outcome type: mixed if {r_evid} contains both 0 and 1
        Set η(q) ← 1 if mixed, else η_homo
        Scale advantages: Ã_{i,t} ← η(q) · Â_{i,t}
    end for
    Update policy: π_θ^(t) ← arg max_θ J^(t)(θ)   ▷ GRPO step
    for each mixed-outcome query q do
        Pick correct rollouts o^+ and incorrect ones o^− from its group
        Extract hint: h^(t) ← T(o^+, o^−)
        Augment prompt: x_q^(t+1) ← x_q^(t) ⊕ h^(t)
    end for
    For homogeneous queries, x_q^(t+1) ← x_q^(t)
end for
return π_θ^(T)
```

A Details of the SHPO Algorithm

Algorithm 1 presents the overall procedure of Self-reflective Hint Policy Optimization (SHPO). Starting from an SFT-initialized locator policy $\pi_\theta^{(0)}$, SHPO iteratively performs rollout sampling, policy optimization, and self-reflective hint augmentation. For each training query q, the current policy samples a group of G reasoning rollouts conditioned on the prompt $x_q^{(t)}$. We then compute token-level rewards and group-normalized advantages following the GRPO framework.

To emphasize informative training signals, SHPO distinguishes between *mixed-outcome* queries and *homogeneous* queries. A query is considered mixed-outcome if its rollout group contains both successful and failed evidence localization trajectories. Such cases indicate that the model already possesses partial reasoning capability but remains unstable. Accordingly, we preserve their original advantages, while down-weighting homogeneous queries using a scaling factor η_{homo}. The policy is subsequently updated using the reweighted advantages through a GRPO optimization step.

After each policy update, SHPO performs self-reflective hint extraction on mixed-outcome queries. Specifically, we compare correct rollouts o^+ against incorrect rollouts o^- within the same query group and employ a teacher model $\mathcal{T}(\cdot)$ to identify the reasoning divergence that caused failure. The teacher model then generates a concise and domain-general corrective hint, which is appended to the original prompt for the next training stage. In contrast, prompts corresponding to homogeneous queries remain unchanged. Through this iterative process, SHPO progressively improves the model's ability to localize evidence by leveraging self-generated reflective guidance derived from its own reasoning trajectories.

B Datasets

Table 6: Dataset statistics. 'Avg.' denotes the average per conversation.

Dataset	LME-S	LME-M	LoCoMo	Long-MT-Bench+
Total Conversations	500	500	10	11
Avg. Sessions	50.2	501.9	27.2	4.9
Avg. Queries	1.0	1.0	198.6	26.2
Avg. Tokens	103,137	1,019,117	20,079	19,195
Session Dates	✓	✓	✓	✗
Retrieval Ground-Truth	✓	✓	✓	✗
QA Ground-Truth	✓	✓	✓	✓
Conversation Type	User-AI	User-AI	User-User	User-AI

Table 6 summarizes the statistics of all datasets, including conversation scale, session depth, and average context length.

LongMemEval-S and **LongMemEval-M** are multi-session User–AI conversation benchmarks designed for long-term memory reasoning and retrieval, where each query requires aggregating relevant information across multiple historical sessions with varying granularity and temporal distance.

LoCoMo is a User–User conversational dataset with long dialogues, where each query involves fine-grained reasoning over a large number of utterances, making it particularly challenging for retrieval and evidence localization.

Long-MT-Bench+ is an extended version of MT-Bench (Zheng et al., 2023b) tailored for long-context conversational evaluation, focusing on multi-turn instruction-following and reasoning under extended dialogue history.

C Retrieval Performance

Table 7 reports retrieval results on LongMemEval-S, LongMemEval-M, and LoCoMo. Across all benchmarks, MemLoc consistently outperforms SeCom and MemGAS, demonstrating the effectiveness and generalizability of its multi-granularity memory retrieval framework. In particular, the

Table 7: Retrieval performance. Full combines MMR and REL; "w/o" removes the specified component. R@k: Recall@k; N@k: NDCG@k.

Method	R@3	N@3	R@5	N@5	R@10	N@10
LongMemEval-S						
SeCom	71.06	80.88	80.43	83.08	89.15	85.11
MemGAS	78.51	**86.83**	88.94	**88.77**	94.47	**89.96**
MemLoc (MMR)	83.90	81.02	90.35	83.38	94.50	84.02
MemLoc (Full)	**87.97**	86.81	**92.56**	88.62	**94.50**	89.44
w/o IM-Routing	86.02	83.36	89.98	85.09	92.60	87.72
w/o CM-Prop.	85.27	83.65	89.23	84.92	92.95	85.55
only SFT	86.45	82.40	92.53	84.77	94.50	85.57
only RL	87.25	86.37	92.08	88.08	94.50	89.03
LongMemEval-M						
SeCom	44.26	56.61	55.32	60.76	66.60	63.78
MemGAS	51.06	61.36	63.62	**66.07**	**77.02**	**69.46**
MemLoc (MMR)	55.92	60.70	65.61	63.65	72.16	65.04
MemLoc (Full)	**63.41**	**61.58**	**69.01**	63.75	72.16	65.01
w/o IM-Routing	62.53	60.48	67.91	63.33	70.35	64.49
w/o CM-Prop.	61.92	60.70	67.61	63.65	70.97	64.04
only SFT	63.30	62.21	68.97	64.41	72.16	65.70
only RL	60.18	58.95	66.55	61.70	72.16	63.95
LoCoMo						
SeCom	55.24	57.90	64.80	62.36	78.30	66.97
MemGAS	57.30	58.76	67.32	63.62	81.82	68.42
MemLoc (MMR)	66.84	54.58	77.10	58.75	90.09	63.83
MemLoc (Full)	**77.85**	**69.76**	**81.72**	**71.67**	**90.09**	**73.80**
w/o IM-Routing	61.61	53.15	72.87	57.88	81.10	62.88
w/o CM-Prop.	63.57	56.00	75.20	60.81	85.70	65.18
only SFT	77.16	69.66	81.58	71.40	90.09	73.58
only RL	64.88	58.74	75.10	62.97	90.09	67.23

full model achieves substantial gains on challenging multi-session datasets such as LongMemEval-M and LoCoMo, indicating that jointly modeling inner-memory Routing and cross-memory propagation is crucial for retrieving dispersed evidence over long conversational histories.

Removing either inner-memory routing (w/o IM-Routing) or cross-memory propagation (w/o CM-Prop.) consistently degrades retrieval performance across datasets, confirming that both modules contribute complementary capabilities. Specifically, IM-Routing improves fine-grained evidence localization within individual memories, while CM-Propagation enhances cross-memory evidence aggregation and long-range contextual propagation. The degradation is particularly pronounced on LoCoMo, where evidence is highly scattered across long interactions, highlighting the necessity of both mechanisms for robust long-context retrieval.

When augmented with REL as an evidence-aware reranker, MemLoc (Full) consistently improves over the MMR-only variant, especially on R@3 and N@3, demonstrating that fine-grained evidence localization effectively refines the retrieval ranking. Within REL, using only SFT or only RL generally underperforms the complete two-stage optimization strategy. This further validates the importance of combining supervised cold-start training with Self-reflective Hint Policy Optimization (SHPO), where SFT provides a stable initialization, while RL-based reflective optimization further enhances evidence integration and improves ranking consistency.

D Ablation of Multi-granularity Memory Retrieval

Table 8: Ablation study of multi-granularity memory retrieval on LongMemEval-S. (✓: enabled, ✗: disabled).

Variant	Sess.	Turn	Event	Summ.	Key.	Time	R@3
Full Method (Ours)	✓	✓	✓	✓	✓	✓	**87.97**
A. Single-Granularity Baselines							
Session Only	✓	✗	✗	✗	✗	✗	76.18
Turn Only	✗	✓	✗	✗	✗	✗	83.34
Event Only	✗	✗	✓	✗	✗	✗	85.15
Summary Only	✗	✗	✗	✓	✗	✗	82.94
Keyword Only	✗	✗	✗	✗	✓	✗	80.21
Time Only	✗	✗	✗	✗	✗	✓	6.44
B. Complementary Granularity Pairs							
Session + Turn	✓	✓	✗	✗	✗	✗	80.05
Session + Event	✓	✗	✓	✗	✗	✗	85.47
Session + Summary	✓	✗	✗	✓	✗	✗	83.72
Turn + Event	✗	✓	✓	✗	✗	✗	86.00
Event + Summary	✗	✗	✓	✓	✗	✗	85.90
C. Progressive Component Addition							
Turn + Event	✗	✓	✓	✗	✗	✗	86.00
+ Session	✓	✓	✓	✗	✗	✗	85.65
+ Summary	✓	✓	✓	✓	✗	✗	87.55
+ Keyword	✓	✓	✓	✗	✓	✗	86.10
D. Ablation of Specific Components							
Full w/o Session	✗	✓	✓	✓	✓	✓	86.07
Full w/o Turn	✓	✗	✓	✓	✓	✓	86.67
Full w/o Event	✓	✓	✗	✓	✓	✓	85.37
Full w/o Time	✓	✓	✓	✓	✓	✗	87.60

We conduct an ablation study on LongMemEval-S to analyze the contribution of different memory granularities. As shown in Table 8, among all single-granularity baselines (*Variant A*), event-level memory (**Event Only**) achieves the best performance, underscoring its central role in long-term memory retrieval. In contrast, session-level memory (**Session Only**) performs substantially worse than finer-grained representations, indicating that excessive noise from overly broad contexts severely degrades retrieval accuracy. Notably, time-level memory (**Time Only**) is insufficient for accurate retrieval, revealing that temporal signals alone lack semantic grounding and become effective only when combined with other memory granularities. However, this result does not imply that temporal modeling is ineffective. Instead, temporal cues such as "last week" "last month" and "yesterday" can effectively narrow the search space to a manageable set of candidate sessions, serving as useful constraints when combined with semantically

grounded memory representations.

Furthermore, our examination of complementary granularity combinations (*Variant B & C*) shows that different memory granularities provide complementary retrieval signals. In particular, combining turn-level and event-level memories yields strong performance, suggesting that fine-grained conversational details and structured event abstractions jointly improve evidence localization.

Finally, removing individual components from the full model (*Variant D*) consistently degrades performance. Notably, removing event-level memory leads to the largest drop, confirming that event-centric memory organization is critical for modeling cross-memory semantic associations and tracking event evolution across long dialogue histories. Although temporal information contributes relatively smaller gains, it remains beneficial for resolving time-sensitive and temporally conflicting queries. Overall, these results validate the effectiveness of our multi-granularity retrieval framework, where different memory granularities collaboratively provide complementary signals for coarse-to-fine evidence localization and cross-memory reasoning.

E Comparison of Different Query Types

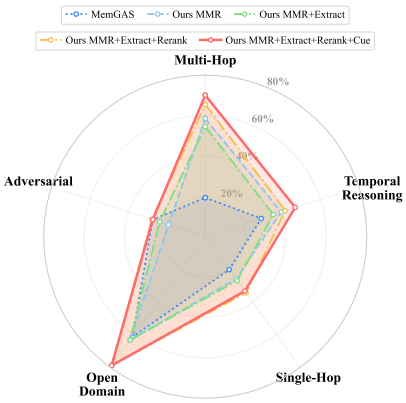

Figure 4: Performance Comparison by Query Type.

To comprehensively evaluate our method's generation performance across different query types, we conduct a detailed query-type analysis on the LoCoMo dataset. As shown in Fig. 4, MemLoc consistently outperforms baselines across all query types. Our MMR variant nearly surpasses MemGAS, demonstrating that our memory retrieval mechanism effectively leverages event associations to identify relevant memory units, thereby enhancing multi-hop reasoning and knowledge integration. Furthermore, the full MMR+REL (Extract-Rerank-Cue) configuration achieves the best results across all query types, confirming that the trained Locator successfully reranks candidate memory units and extracts critical evidence, leading to more accurate and reliable answer generation.

F Implementation Details

For each query, we retrieve the Top-10 sessions via *Inner-Memory Routing* and *Cross-Memory Propagation*, apply Locator for *Inner-Memory Extraction* and *Cross-Memory Reranking*, and use the reranked Top-3 sessions for answer generation. We employ **BGE-M3** (Chen et al., 2023) as the dense retriever and **GPT-4o mini** as the backbone for all generation and extraction tasks. All baselines use identical prompts with temperature 0, and every experiment is run three times in a zero-shot setting. For *Inner-Memory Routing*, we use a softmax temperature of $\tau = 0.1$. For *Memory Graph Construction*, we link each node to its top-$K_{\text{cand}} = 20$ most similar same-granularity units, fit a two-component GMM to their similarity scores, and retain only neighbors in the higher-mean component. For *Cross-Memory Propagation*, we sparsify the seed relevance vector to its top-$K_0 = 15$ entries and set the PPR restart probability to $d = 0.1$.

Training. Locator is trained in two stages.

Stage 1 (SFT cold-start): We fine-tune for 5 epochs with learning rate 10^{-5} on 4,000 multi-hop QA samples (2–4 hops) from HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), and 2WikiMultihopQA (Ho et al., 2020).

Stage 2 (SHPO): We adapt 6,000 conversational QA samples from Time-Dialogue (Du et al., 2025b; Qin et al., 2021) and HaluMem (Chen et al., 2026) to our memory-based QA. Training uses a 3-stage GRPO (60 steps per stage, learning rate 10^{-6}), sampling 4 rollouts per query at each step. Queries with mixed outcomes (both correct and incorrect rollouts) yield contrastive hints, which are appended to the prompts of subsequent stages. We set the KL coefficient $\gamma = 0.01$ and the reward weights $\alpha = 0.3$ and $\beta = 0.6$.

Implementation. All experiments are built with Verl (Zheng et al., 2025; Sheng et al., 2024) and LlamaFactory (Zheng et al., 2024) on 8×A800 GPUs (PyTorch). *Long-MT-Bench+* is excluded from retrieval evaluation due to missing ground-truth annotations. For *LoCoMo*, images are replaced with

Table 9: Retrieval performance with different encoders on the LoCoMo dataset.

Model	Recall@3	NDCG@3	Recall@5	NDCG@5	Recall@10	NDCG@10
Base Retriever: MiniLM						
MiniLM	42.55	44.19	52.37	49.01	67.98	54.59
RecurSum	44.76	46.82	54.73	51.64	72.16	57.52
SeCom	45.77	47.25	54.93	51.70	71.15	57.37
MemGAS	47.73	49.11	56.60	53.46	71.30	58.59
MemLoc (Ours w/o Locator)	**53.19**	42.16	**64.76**	46.90	**80.79**	52.61
Base Retriever: MPNet						
MPNet	45.92	47.71	53.98	51.79	68.58	56.88
RecurSum	49.50	51.15	59.47	56.16	76.64	61.99
SeCom	47.53	49.03	57.05	53.57	70.90	58.57
MemGAS	52.77	54.63	62.79	59.56	80.51	65.48
MemLoc (Ours w/o Locator)	**56.77**	43.82	**67.80**	48.35	**82.96**	53.87
Base Retriever: Contriever						
Contriever	49.90	52.15	58.26	56.29	71.80	60.92
RecurSum	47.23	48.99	59.01	54.58	74.97	60.07
SeCom	55.24	57.90	64.80	62.36	78.30	66.97
MemGAS	57.30	58.76	67.32	63.62	81.82	68.42
MemLoc (Ours w/o Locator)	**64.56**	51.21	**74.42**	55.32	**88.57**	60.54
Base Retriever: BGE-M3						
BGE-M3	51.86	47.45	60.66	51.08	75.54	56.18
RecurSum	58.96	52.94	69.71	57.38	84.35	62.44
SeCom	56.11	49.75	66.59	54.11	82.27	59.50
MemGAS	60.62	52.60	71.46	57.14	86.07	62.22
MemLoc (Ours w/o Locator)	**66.84**	54.58	**77.10**	58.75	**90.09**	63.83

textual captions. Results for *RAPTOR*, *A-Mem*, and *HippoRAG 2* on *LongMemEval-M* are unavailable due to excessive runtime.

G Generalization Experiments

G.1 Retrieval Generalization

As shown in Table 9, MemLoc consistently improves retrieval performance across all base encoders, including MiniLM, MPNet, Contriever, and BGE-M3 on the LoCoMo dataset. Importantly, the gains are stable across both lightweight and strong retrievers, indicating that our method is not dependent on any specific embedding model, but instead serves as a plug-and-play enhancement for diverse retrieval backbones. In particular, MemLoc improves Recall@K across all settings, demonstrating its ability to refine retrieval signals beyond the capacity of the underlying encoders. These results demonstrate the encoder-agnostic nature and strong generalization ability of MemLoc across heterogeneous embedding backbones.

G.2 Generation Generalization

We further evaluate the generalization of MemLoc across different generation backbones, including GPT-4o mini, Qwen3-30B-A3B, and Qwen3-8B under both No-Think and Think settings. As shown in Table 10, MemLoc consistently improves QA performance over MemGAS across all datasets and backbone models, demonstrating strong cross-backbone robustness. Notably, the improvements are observed for both strong and weak generators, indicating that MemLoc does not rely on the reasoning capacity of the stronger generator, but instead benefits from higher-quality retrieved evidence for downstream generation.

H Efficiency and Latency Analysis

As shown in Table 11, MemLoc achieves the highest LLM-as-a-Judge (4o-J) evaluation scores across all datasets while consuming substantially fewer tokens than the Full History baseline. Compared to the efficient MemGAS baseline, MemLoc incurs only a marginal increase in token usage and inference latency, while consistently achieving notable gains in LLM-as-a-Judge scores. These results

Table 10: QA performance under the same Top-3 retrieval setting. Best results are shown in **bold**.

Method	Generator	*LongMemEval-S*				*LongMemEval-M*			
		4o-J	F1	B-4	BS	4o-J	F1	B-4	BS
MemGAS	GPT-4o mini	60.20	20.38	**4.22**	**85.21**	45.40	16.85	**3.39**	**84.69**
	Qwen3-30B-A3B	58.40	**26.65**	**6.10**	**86.61**	47.80	**25.25**	**5.24**	**86.26**
	Qwen3-8B (No-Think)	43.20	18.75	3.77	85.57	39.40	18.97	3.77	85.57
	Qwen3-8B (Think)	67.20	12.86	**2.20**	83.21	54.20	12.26	1.77	83.01
MemLoc (Ours)	GPT-4o mini	**68.40**	**21.52**	3.81	84.33	**54.20**	**17.27**	3.08	83.71
	Qwen3-30B-A3B	**71.20**	17.72	3.51	84.74	**55.60**	15.88	2.83	84.27
	Qwen3-8B (No-Think)	**58.40**	**19.54**	**3.92**	**85.34**	**48.60**	18.20	3.33	84.99
	Qwen3-8B (Think)	**73.00**	**13.06**	2.12	**83.64**	**58.20**	**12.38**	**1.89**	**83.38**

Method	Generator	*LoCoMo*				*Long-MT-Bench+*			
		4o-J	F1	B-4	BS	4o-J	F1	B-4	BS
MemGAS	GPT-4o mini	41.07	17.66	3.61	85.13	69.44	41.49	**15.62**	**88.96**
	Qwen3-30B-A3B	43.45	**24.41**	**4.31**	**87.04**	63.89	**54.70**	13.38	**89.00**
	Qwen3-8B (No-Think)	47.78	20.18	**5.36**	**87.69**	54.51	**53.60**	13.16	88.95
	Qwen3-8B (Think)	48.33	**12.93**	**2.32**	84.64	67.36	38.32	6.87	85.70
MemLoc (Ours)	GPT-4o mini	**57.40**	**19.99**	**4.26**	**85.26**	**77.42**	**46.65**	14.02	88.48
	Qwen3-30B-A3B	**52.82**	17.89	3.68	86.37	**73.26**	48.47	**13.94**	88.51
	Qwen3-8B (No-Think)	**54.92**	**23.25**	4.80	87.16	**70.13**	50.18	**15.07**	**89.02**
	Qwen3-8B (Think)	**53.82**	12.62	2.18	**85.21**	**69.79**	**40.10**	**11.30**	**87.96**

Table 11: Efficiency and Latency comparison.

Dataset	Method	Tokens	Filtering (s)	Generation (s)	4o-J
LongMemEval-M	Full History	128,000	0.00	12.88	12.20
	MemGAS	8,852	0.53	1.48	45.40
	MemLoc	8,537	0.66	1.83	**54.20**
LongMemEval-S	Full History	103,137	0.00	9.39	50.60
	MemGAS	8,829	0.54	0.94	60.20
	MemLoc	8,941	0.61	1.58	**68.40**
LoCoMo	Full History	20,079	0.00	4.92	33.43
	MemGAS	2,825	0.47	1.48	41.07
	MemLoc	2,907	0.46	1.51	**57.40**
Long-MT-Bench+	Full History	19,195	0.00	4.72	67.44
	MemGAS	12,873	0.73	1.60	69.44
	MemLoc	13,921	0.31	3.28	**77.42**

Tokens: Average token consumption per query.
Filtering: Average filtering latency per session.
Generation: Average answer generation latency per query.
4o-J: Measuring response quality (higher is better).

Table 12: Offline indexing complexity and QA performance on LongMemEval-M ($S \approx 500$, $N > 250$K). "OOT": Out-Of-Time.

Method	Memory Extraction	Graph/Index Build	4o-J
Full History	–	–	12.20
RAPTOR	$O(N \cdot C_{\text{LLM}})$	$O(N^2)$	OOT
A-Mem	$O(N \cdot C_{\text{LLM}})$	$O(N^2 + N \cdot C_{\text{LLM}})$	OOT
HippoRAG 2	$O(N \cdot C_{\text{LLM}})$	$O(N^2)$	OOT
Mem0	$O(N \cdot C_{\text{LLM}})$	$O(N^2)$	32.00
MemoryTree	$O(N \cdot C_{\text{LLM}})$	$O(N \log N)$	35.80
Memory-R1	$O(N \cdot C_{\text{LLM}})$	-	29.00
SeCom	$O(S \cdot C_{\text{LLM}})$	-	42.80
RecurSum	$O(S \cdot C_{\text{LLM}})$	-	23.80
MemoryAgent	$O(S \cdot C_{\text{LLM}})$	–	34.20
MemSkill	$O(S \cdot C_{\text{LLM}})$	-	42.20
MemGAS	$O(S \cdot C_{\text{LLM}})$	$O(G \cdot S^2)$	45.40
MemLoc (Ours)	$O(S \cdot C_{\text{LLM}})$	$O(G \cdot S^2)$	**54.20**

suggest that our multi-granularity retrieval and evidence localization pipeline effectively compresses long-context inputs by filtering irrelevant content and retaining relevant information, thereby achieving state-of-the-art performance with low computational overhead.

H.1 Computational Cost Analysis

Table 12 compares the offline indexing complexity and QA performance of different memory systems on LongMemEval-M. We decompose the offline cost into two stages: memory extraction (node construction) and graph/index construction (edge construction). Let S denote the number of dialogue sessions, $\bar{n}$ the average number of utterances per session, and $N = S \cdot \bar{n}$ the total number of utterances. We further denote by G the number of memory granularities, and C_{LLM} the cost of a single LLM inference call.

Table 13: Best results are shown in **bold**, and second-best results are underlined. **4o-J** refers to GPT-4o-as-Judge, **B-4** to BLEU-4, and **BS** to BERTScore. **Tokens** denotes the average token consumption per query. All methods use the same generator GPT-4o mini.

Model	*LongMemEval-S*					*LongMemEval-M*				
	4o-J	F1	B-4	BS	Tokens	4o-J	F1	B-4	BS	Tokens
Full History	50.60	11.48	1.40	83.07	103,137	12.20	5.70	0.78	81.62	128,000
RAPTOR	32.20	12.08	1.90	83.50	6,254	*Timeout*				
SeCom	56.00	12.95	2.25	83.51	2,741	42.80	11.33	1.79	83.36	2,821
A-Mem	55.60	13.73	2.11	83.88	9,018	*Timeout*				
HippoRAG 2	57.60	14.73	2.15	83.86	8,530	*Timeout*				
Mem0	42.00	17.72	3.62	83.25	6,787	32.00	14.38	2.57	82.34	7,452
MemGAS	60.20	20.38	4.22	85.21	8,829	45.40	16.85	**3.39**	84.69	8,852
MemoryAgent	42.80	11.71	3.91	81.48	11,200	34.20	9.58	2.90	81.33	11,800
Memory-R1	45.60	21.49	**8.66**	**87.06**	13,500	29.00	5.95	1.06	80.55	14,200
RecurSum	35.40	12.29	2.09	83.60	8,853	23.80	10.04	1.70	83.12	8,927
MemoryTree	41.40	12.02	2.45	82.90	7,600	35.80	11.57	2.23	83.08	8,200
MemSkill	59.14	15.61	3.66	84.35	6,900	42.20	14.88	3.42	84.07	7,300
MemLoc (Ours, 4B)	65.80	16.80	3.78	85.40	8,645	48.20	13.41	2.90	84.71	8,326
MemLoc (Ours, 8B)	**68.40**	**21.52**	3.81	84.33	8,941	**54.20**	**17.27**	3.08	83.71	8,537

Model	*LoCoMo*					*Long-MT-Bench+*				
	4o-J	F1	B-4	BS	Tokens	4o-J	F1	B-4	BS	Tokens
Full History	33.43	12.23	1.84	84.07	20,079	67.44	36.07	11.32	87.81	19,195
RAPTOR	31.72	14.55	2.88	84.48	1,931	59.72	37.69	13.47	88.38	10,631
SeCom	44.21	13.79	2.30	84.04	1,021	64.58	36.68	12.01	87.88	4,714
A-Mem	40.81	14.72	2.83	84.72	3,042	65.73	36.82	11.36	87.92	13,735
HippoRAG 2	45.62	16.66	2.91	84.88	2,991	63.54	35.64	11.05	87.70	13,583
Mem0	36.39	8.16	2.06	83.70	2,450	38.33	25.42	6.06	86.86	8,100
MemGAS	41.07	17.66	3.61	85.13	2,825	69.44	41.49	**15.62**	**88.96**	12,873
MemoryAgent	38.72	14.80	**4.76**	81.75	6,800	39.03	11.69	2.06	82.48	16,200
Memory-R1	38.07	5.52	1.01	81.32	7,900	55.90	8.46	1.41	81.37	18,400
RecurSum	22.56	9.14	0.99	83.45	3,074	24.65	26.58	6.91	86.11	13,527
MemoryTree	35.21	8.95	1.62	83.49	5,300	51.39	30.13	8.55	86.99	11,500
MemSkill	47.21	17.37	3.37	85.43	4,900	62.33	32.01	9.89	87.44	10,700
MemLoc (Ours, 4B)	57.20	18.13	4.13	**86.88**	2,744	73.96	37.98	13.70	88.49	12,463
MemLoc (Ours, 8B)	**57.40**	**19.99**	4.26	85.26	2,907	**77.42**	**46.65**	14.02	88.48	13,921

Analysis. (1) Session-level extraction substantially reduces offline indexing cost. Methods such as RAPTOR, A-Mem, HippoRAG 2, Mem0, MemoryTree, and Memory-R1 perform memory extraction at the utterance level, resulting in $O(N \cdot C_{\text{LLM}})$ complexity. On LongMemEval-M, where N>250K, this requires a prohibitively large number of LLM calls, causing several methods to exceed the runtime budget. In contrast, MemLoc performs joint multi-granularity extraction with a single LLM call per session, reducing the extraction cost to $O(S \cdot C_{\text{LLM}})$. **(2) Graph construction introduces moderate overhead but yields substantial gains.** Both MemGAS and MemLoc construct cross-session relational graphs with complexity $O(G \cdot S^2)$ based on embedding similarity. Unlike methods that require LLM inference during graph construction, our pipeline operates entirely on pre-computed embeddings. Although graph construction incurs additional offline computation compared with flat retrieval systems such as SeCom and RecurSum, the resulting relational structure enables effective cross-memory propagation and multi-hop evidence association across sessions. **(3) Structured graph indexing substantially improves QA quality.** Flat retrieval methods without explicit relational modeling achieve limited performance despite lower indexing cost. In contrast, MemLoc achieves the best overall QA performance while maintaining computational complexity comparable to existing graph-based methods. These results demonstrate that incorporating structured cross-memory graph construction enables more effective evidence association and long-context reasoning, yielding substantial gains in long-term conversation QA quality with only moderate offline overhead.

I LLM-as-Judge and Human Evaluation

We use GPT-4o as the primary judge, following prior work including MemSkill, MemGAS, and Memory-R1. LLM-based evaluation is particularly

important for long-term memory QA, where semantically equivalent answers can differ substantially in surface form. For example, in LoCoMo, the question "How many doctor's appointments did I go to in March?" has the ground-truth answer "2", while our model produces "Two doctors, the first is dentist Tom, the second is dermatologist Mike." Although semantically correct, the token-level F1 score is only 7.7. This discrepancy is further supported by the BERTScore results in Appendix G (Table 13), where high semantic similarity can coexist with low lexical overlap.

Table 14: LLM-as-judge evaluation on LongMemEval-S using different judges. Scores are percentages.

Method	GPT-4o	GPT-5	Claude-Sonnet 4.6	DeepSeek-V4-Pro
MemLoc (Ours)	**68.40**	**69.60**	**70.00**	**66.80**
MemGAS	60.20	60.80	60.00	58.20
MemSkill	59.14	55.80	56.40	54.60
Memory-R1	45.60	41.20	41.80	41.20
MemoryAgent	42.80	41.80	43.40	39.80
Mem0	42.00	39.80	40.40	38.40
MemoryTree	41.40	42.60	43.20	40.40

To examine potential judge bias, we additionally evaluate LongMemEval-S responses using GPT-5, Claude-Sonnet 4.6, and DeepSeek-V4-Pro. As shown in Table 14, MemLoc consistently ranks first across all four judges, with an 8–10 point margin over the strongest baseline. The variation across judges is also small, suggesting that the observed improvements are robust to the choice of evaluator.

We further conduct human evaluation on all 500 LongMemEval-S responses. Two annotators independently assess each response, with a third annotator resolving disagreements. The resulting Cohen's $\kappa = 0.92$ indicates almost perfect inter-annotator agreement. Human evaluation is consistent with the LLM-based results and confirms that MemLoc significantly outperforms the compared baselines.

J Prompt Templates

We provide the prompt templates used at different stages of our framework. In the templates, blue text denotes sample-specific inputs, while black text represents fixed instructions shared across all instances. Since our framework is model-agnostic, these prompts can be freely integrated with different LLMs and readily benefit from future advances in foundation models.

Prompt Template for Summarization and Keyword Extraction

Instruction: You are an intelligent and insightful individual. Your task is to analyze the conversation between a user and an AI assistant and extract two key elements: a concise summary of the conversation and the most relevant keywords.

Input: {Conversation between User and AI Assistant}

Tasks:
1. **Summary**: Provide a concise paragraph that summarizes the main topics and key information of the conversation.
2. **Keywords**: Extract the most relevant keywords from the conversation content.

Output Format:
Return a JSON object that must strictly contain the following structure:

```
{
    "memory":
        {
        "summary": "<A concise summary of the conversation>",
        "keywords": "<Keyword 1>; <Keyword 2>; <Keyword 3>; ..."
        }
}
```

Only provide the JSON object without any additional text.

Answer:

Prompt Template for Event Extraction and Timeline Construction

Instruction: You are an intelligent system designed to extract structured temporal information from conversations. Your task is to identify key events and construct a standardized timeline based on the given dialogue.

Input:
Conversation Date: {Conversation Date}
Conversation Content: {Conversation between User and AI Assistant}

Tasks:
1. **Key Event Extraction**: Identify the key events mentioned in the conversation.
2. **Timeline Construction**: Organize the extracted events in chronological order.
3. **Role Awareness**: Summarize the information provided by the **assistant** and the **user** separately.

Requirements:
- Output only concise event summaries.
- Do NOT quote or restate the original dialogue.
- Ensure the results are clear, structured, and information-dense.

Output Format:

```
| Date | Key Event | Description |
|------|-----------|-------------|
| YYYY/MM/DD | xxx | xxx |
| YYYY/MM/DD | xxx | xxx |
```

Answer:

Prompt Template for Evidence Filtering and Sentence Selection

Instruction: You are an expert in information retrieval and evidence filtering. Your task is to identify the sentences that are most relevant to answering a given question based on retrieved documents.

Input:
Question: {Question}
Retrieved Documents: {List of sentences with IDs}

Tasks:
1. Analyze the relevance of each sentence to the question.
2. Select the most relevant sentences.
3. Provide a final answer based on the selected evidence.

Requirements:
- Perform step-by-step reasoning.
- Only select the most relevant sentences.
- Ensure the final answer is accurate and grounded in the selected evidence.

Output Format:

```
<reason>...step-by-step reasoning...</reason>
<id>Most relevant sentence IDs</id>
<answer>answer</answer>
```

Answer:

Prompt Template for Multi-Granular Reasoning and Answer Generation

Instruction: You are an intelligent conversational assistant. Your task is to carefully analyze the provided conversation history and multi-granular contextual information, and generate a concise, accurate, coherent, and helpful answer to the given question.

Input:
Conversation History and Context: {Retrieved texts with global IDs}
Reference Information (optional): {Global ID mapping}
Question: {Question}

Tasks:
1. Understand the conversation history and contextual signals.
2. Integrate multi-granular information (e.g., session, summary, events, time).
3. Generate a coherent and accurate answer.

Requirements:
- The answer must be grounded primarily in the provided conversation history.
- Do NOT introduce unsupported or external information.
- Keep the answer concise and focused.
- Ensure logical coherence and readability.
- If the question involves time, provide explicit timestamps when possible.
- Prefer reusing original wording from the conversation when appropriate.
- Use reference information only as supporting clues, not as the sole basis.
- Answer must be in English.

Output Format:
Provide a direct and concise answer to the question.

Answer:

Prompt Template for Answer Verification

Instruction: Determine whether the model response correctly answers the question based on the reference answer.

Input:
Question: {Question}
Reference Answer: {Answer}
Model Response: {Response}

Requirement:
Output `[[yes]]` if the response is correct or equivalent to the reference answer; otherwise output `[[no]]`.

Output Format:

```
[[yes]] or [[no]]
```

Answer:

Prompt Template for SFT & RL Training

Instruction: You are an expert in information retrieval and evidence filtering. Given a question and a list of retrieved documents, select the documents that are most relevant to answering the question. Output the corresponding document IDs, answer and explain your reasoning.

Input:
Question: {Question}
Retrieved Documents: {Documents}

Output Format:
<reason>...step-by-step reasoning...</reason>
<id>Most relevant document IDs</id>
<answer>answer</answer>

Prompt Template for Hint Extraction

Instruction: You are an intelligent and analytical teacher model. Your task is to analyze the divergence between correct reasoning trajectories and wrong reasoning trajectories, then extract a concise and domain-general corrective hint without leaking the ground-truth answer.

Input:
Question: {Question}
Correct Reasoning Trace: {Correct Reasoning Trace}
Wrong Reasoning Trace: {Wrong Reasoning Trace}
Answer: {Answer}

Tasks:
1. **Identify Divergence**: Determine which reasoning step caused the student model to deviate from the correct reasoning path.
2. **Generate Hint**: Produce a concise, domain-general strategy that can guide the student model back to the correct reasoning direction. The hint should focus on reasoning principles rather than specific answers.
3. **Leakage Verification**: Ensure that the generated hint does not reveal or directly imply the ground-truth answer.

Example Hints:

- When multiple timestamps conflict, prioritize the most recent entry as the user's current state.
- Verify whether the event occurred or was merely planned. Distinguish intent from completion.
- Disregard assistant-generated summaries when they contradict explicit user statements.
- Cross-check person attributes (age, location) across sessions; use the latest update.

Output Format:
Return a JSON object that must strictly contain the following structure:

```
{
    "hint":
    {
        "divergence_step": "<Reasoning step causing divergence>",
        "strategy_hint": "<Domain-general corrective hint>"
    }
}
```

Only provide the JSON object without any additional text.

Answer: