GraP-Mem: Granularity-Aware Planning for Adaptive Memory Access in Long-Horizon LLM Agents

Anonymous ACL submission

Abstract

LLM agents increasingly operate in long-horizon tasks and extended dialogues, where they must retrieve useful past experiences beyond limited context windows. A central challenge is to balance efficient recall with faithful evidence recovery. Existing memory methods often rely on static access strategies: compact memories reduce context cost but may omit fine-grained evidence, whereas full histories preserve details but introduce redundant and noisy context. We propose GraP-Mem, a multi-granularity memory planning framework that adaptively schedules memory access for each question. GraP-Mem represents each historical session with two complementary views: compact semantic memories for efficient retrieval and source contexts for detailed evidence recovery. During inference, a Plan Agent generates retrieval-oriented information needs and queries, while an Integration Agent maintains a question-specific evidence state that consolidates useful evidence and identifies remaining gaps. When the evidence is incomplete, GraP-Mem selectively expands access to source contexts or performs additional retrieval. The final Answer Model is conditioned only on the integrated evidence state. Experiments on LoCoMo and NarrativeQA show that GraP-Mem consistently improves F1 and BLEU across backbone models, demonstrating the effectiveness of plan-guided memory access across multiple granularities.

1 Introduction

The rapid progress of large language models has enabled language agents to reason, plan, and interact with complex environments (Zhang et al., 2025; Chen et al., 2024). As these agents are increasingly deployed in long-horizon tasks and extended dialogues, they must maintain continuity over multi-step interactions, reuse historical experience, and recover task-relevant evidence across sessions (Chen et al., 2025; Wang et al., 2025c). However, LLMs are inherently stateless and constrained by finite context windows (Wang et al., 2024; Fang et al., 2025a). These limitations make it difficult for agents to preserve useful historical information while keeping the active context focused and manageable, motivating the need for effective memory mechanisms (Tang et al., 2025; Ouyang et al., 2025; Fang et al., 2025b).

Recent studies have introduced a variety of memory mechanisms for LLM agents. Some methods store historical information as external memory records and retrieve relevant content through vector indices or embedding-based retrieval modules (Packer et al., 2023; Chhikara et al., 2025; Zhong et al., 2024). Others organize memory by extracting entities, events, or relations from interaction histories and building structured links among memory units (Edge et al., 2024; Li et al., 2025a; Xu et al., 2025; Han et al., 2024). Multi-category memory systems further divide experiences into different types of records to support personalized and long-term interaction (Wang et al., 2025b; Hu et al., 2025). These approaches improve the storage and retrieval of past experiences, but many of them still rely on relatively fixed memory access strategies, applying similar retrieval or compression procedures across tasks with different information needs (Fang et al., 2025a; Hu et al., 2025).

This fixed-access paradigm creates a fundamental trade-off between efficient recall and faithful evidence recovery. Compact or distilled memories reduce storage cost, retrieval noise, and context overhead, but they may omit fine-grained details that are crucial for complex reasoning, temporal reasoning, or evidence-sensitive questions (Fang et al., 2025b; Chhikara et al., 2025). In contrast, preserving complete interaction histories reduces information loss, but it often introduces redundant and task-irrelevant content, increases inference cost, and lowers the density of useful evidence (Li et al., 2025b; Zhong et al., 2024; Hu et al., 2025). Such redundancy can

also worsen middle-context degradation in long-context reasoning (Liu et al., 2024). Some agentic frameworks attempt to improve retrieval relevance through online filtering or iterative reasoning, but repeated inference cycles introduce additional latency and token costs (Yan et al., 2025; Packer et al., 2023). The key limitation, therefore, is not only how memory is stored, but also how agents decide which granularity of memory should be exposed to the reasoning process for a given question.

To address this trade-off, we propose GraP-Mem, a multi-granularity memory planning framework for long-horizon LLM agents. The central idea is to separate memory preservation from memory exposure and adapt memory access to the evidence needs of each question. During memory construction, GraP-Mem represents each historical session with two complementary views: compact semantic memories rewritten by a Memory Agent for efficient coarse retrieval, and source contexts retained for fine-grained evidence recovery. During inference, GraP-Mem treats memory access as a progressive plan–retrieve–integrate process rather than a one-shot similarity search. A Plan Agent analyzes the question and generates retrieval-oriented information needs and queries, while an Integration Agent consolidates retrieved information into a question-specific evidence state and identifies remaining evidence gaps. When the evidence is incomplete, GraP-Mem selectively expands access from compact memories to source contexts and updates the evidence state with newly recovered details. In this way, simple questions can be answered with low-cost compact retrieval, whereas complex questions can recover detailed historical evidence only when necessary.

Our contributions are as follows:

- We propose GraP-Mem, a multi-granularity memory framework that represents each historical session with both compact semantic memory and source context, balancing efficient recall with fine-grained evidence recovery.
- We design a planning-guided progressive retrieval mechanism, where the Plan Agent transforms questions into retrieval plans and adaptively schedules source-context access according to unresolved evidence needs, rather than relying on fixed top-k retrieval or full-history expansion.
- We introduce a query-conditioned evidence integration module that maintains an evolving evidence state, consolidates useful memories, filters redundant context, and supports evidence-grounded answer generation.

2 Related Work

2.1 Memory Systems for LLM Agents

Memory systems extend LLM agents beyond fixed context windows by storing and retrieving historical interactions for long-term use. Existing methods commonly treat memory as an external context resource, where past information is retained and retrieved on demand (Packer et al., 2023; Kang et al., 2025; Wang et al., 2025a). However, raw or lightly processed histories often accumulate redundant and task-irrelevant content over extended interactions. To improve memory organization, recent work extracts entities, events, or relations and stores them as structured records or graphs (Zhong et al., 2024; Chhikara et al., 2025; Edge et al., 2024; Li et al., 2025a; Xu et al., 2025; Han et al., 2024), while other systems introduce multiple memory categories or routing mechanisms for personalized and fine-grained access (Wang et al., 2025b; Hu et al., 2025). Although these designs improve memory representation, many still rely on fixed inference-time access patterns, such as similarity-based retrieval or predefined routing rules. In contrast, GraP-Mem focuses on adaptive memory access: it combines compact semantic memories with complete source contexts and uses a Plan Agent to guide progressive retrieval according to query-specific information needs, enabling lightweight recall for simple questions and detailed evidence recovery for complex ones.

2.2 Context Management

Context management determines which historical information should be exposed to LLM agents during inference. Long-context models and context extension techniques allow more history to be included in prompts, but they increase computation cost and can suffer from the lost-in-the-middle problem (Liu et al., 2024). RAG-based methods construct query-specific contexts by retrieving relevant passages from external sources (Lewis et al., 2020; Asai et al., 2024; Jiang et al., 2023), with structured variants further improving retrieval through graph-based or lightweight indexing (Edge et al., 2024; Guo et al., 2024). However, these methods are mainly designed for static knowledge access, whereas agent memory is continuously up-

dated, temporally grounded, and tied to interaction-specific experience. Other approaches improve token efficiency through prompt compression, memory filtering, or summarized memory units (Fang et al., 2025b; Chhikara et al., 2025; Xu et al., 2025), but compressed contexts may lose fine-grained evidence, while raw-history contexts often introduce redundancy and unnecessary inference cost (Yan et al., 2025). GraP-Mem addresses this trade-off by treating context construction as adaptive retrieval over multi-granularity memories: it first constructs a compact evidence context from semantic memories and selectively accesses complete source contexts only when additional evidence is required.

3 Preliminary

We formalize the task setting for memory agents. For each input sample, the agent observes a sequence of historical sessions $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$, where each session records a coherent segment of past interaction, such as a dialogue episode, a task trajectory, or a narrative context. The agent processes these sessions sequentially and maintains a memory state $\mathcal{M}^{(t)}$ after observing the first t sessions. Starting from an initial memory $\mathcal{M}^{(0)}$, the update process is defined as

$$\mathcal{M}^{(t)} = \Phi(s_t, \mathcal{M}^{(t-1)}), \tag{1}$$

where Φ denotes the memory construction process. After all sessions are processed, $\mathcal{M}^{(N)}$ serves as the memory store for future queries.

Given a query $q \in \mathcal{Q}$, the agent should construct query-relevant evidence from memory rather than expose the entire history to the model. We denote this memory access process as

$$\mathcal{E}_q = \Psi(q, \mathcal{M}^{(N)}), \tag{2}$$

where $\mathcal{E}_q$ is the evidence state constructed for answering q. This evidence state may contain selected, organized, or compressed information from memory. The final output is generated by

$$y = F(q, \mathcal{E}_q), \tag{3}$$

where F denotes the answer generation function. Under this formulation, the goal is to design Φ and Ψ to preserve useful historical information, avoid unnecessary context exposure, and provide sufficient evidence for reliable generation across queries with different information needs.

4 Method

4.1 Overview

GraP-Mem is a multi-granularity memory planning framework for long-horizon LLM agents. As shown in Figure 1, each historical session is represented with two complementary views: a compact semantic memory document for efficient retrieval and a source context for detailed evidence recovery. This design separates memory preservation from memory exposure: source contexts retain complete historical information, while compact memory documents provide the default retrieval view.

Given a question, GraP-Mem follows a plan–retrieve–integrate process. The Plan Agent analyzes the information need and retrieves relevant compact memories, while the Integration Agent consolidates useful information into a question-specific evidence state. When this evidence state reveals remaining gaps, the Plan Agent selectively expands access to source contexts, and the Integration Agent updates the evidence with recovered details. The final Answer Model generates the answer only from the integrated evidence. In this way, GraP-Mem adapts memory access to each question, supporting low-cost recall for simple queries and fine-grained evidence recovery for complex ones.

4.2 Multi-Granularity Memory Construction

A central challenge in long-horizon memory construction is the granularity mismatch between efficient access and evidence completeness. Compact memories support efficient retrieval but may omit fine-grained details, while complete histories preserve information but introduce redundant context during inference. GraP-Mem addresses this issue by constructing two complementary views for each historical session: a compact semantic memory document for efficient retrieval and a source context for detailed evidence recovery.

Given historical sessions $\mathcal{S} = \{s_i\}_{i=1}^{N}$, GraP-Mem preserves each session as its source context:

$$c_i = s_i, \tag{4}$$

where c_i retains the complete session without compression. The compact view is produced by the Memory Agent Φ_{write}, which extracts and rewrites useful information from s_i into a session-level compact semantic memory document:

$$d_i = \Phi_{\text{write}}(s_i) = [u_{i,1}; u_{i,2}; \cdots; u_{i,n_i}]. \tag{5}$$

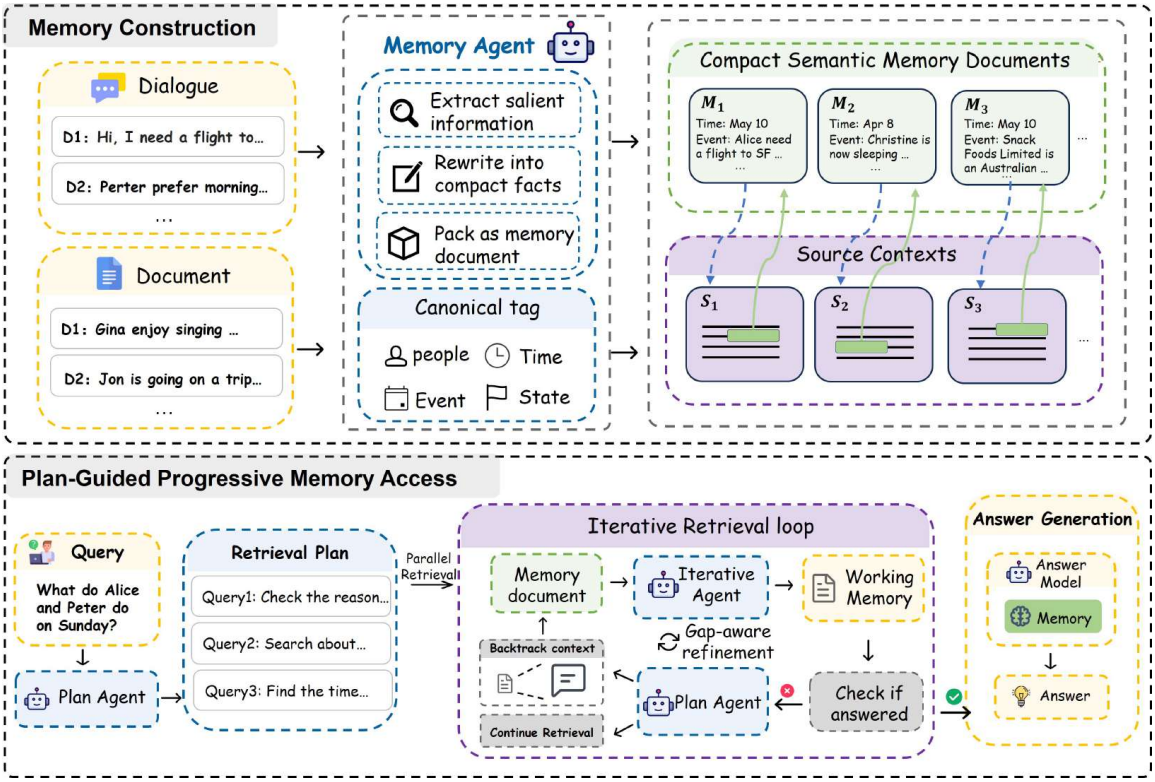

Figure 1: Overview of the proposed GraP-Mem framework.

Here, d_i denotes the compact memory document, $u_{i,j}$ is the j-th self-contained memory record in d_i, and n_i is adaptively determined by the amount of salient information in the session. Each record is written as a compact factual statement that preserves useful evidence, such as salient entities, events, temporal cues, task states, decisions, and evidence-bearing facts, while removing redundant or low-value content.

The document d_i serves as the retrieval unit for session s_i. Rather than uniformly summarizing the whole session, it provides a high-density semantic representation that is easier to match with retrieval queries than the full source context c_i. The resulting multi-granularity memory store is defined as

$$\mathcal{M} = \{(d_i, c_i)\}_{i=1}^{N}. \tag{6}$$

Each pair (d_i, c_i) links the compact retrieval view to the corresponding source context. This construction separates memory preservation from memory exposure: complete histories are preserved in source contexts, while compact semantic memory documents provide the default view for efficient memory access and can be traced back to source contexts when fine-grained evidence is required.

4.3 Plan-Guided Progressive Memory Access

Given the multi-granularity memory store, GraP-Mem needs to decide which memory view should be accessed for each query. A fixed retrieval strategy is often suboptimal: compact semantic memories support efficient recall but may miss fine-grained evidence, whereas source contexts preserve details but can introduce redundant content and unnecessary inference cost. GraP-Mem therefore formulates memory access as a plan-guided progressive retrieval process, starting from compact semantic memories and expanding to source contexts only when additional evidence is required.

Let q be the input question. At retrieval step t, GraP-Mem maintains a question-specific evidence state $\mathcal{E}_t$ and an access history $\mathcal{H}_t$. The evidence state stores useful information collected so far, while the access history records previous access decisions, especially opened source contexts, to reduce unnecessary repeated reading. Based on the question, current evidence, evidence gap, and access history, the Plan Agent generates a retrieval-oriented plan:

$$p_t = \Phi_{\text{plan}}(q, \mathcal{E}_t, g_t, \mathcal{H}_t), \tag{7}$$

where g_t denotes the evidence gap identified after the previous integration step, and $g_0 = \emptyset$ for the

initial retrieval step. We write the plan as

$$p_t = (\mathcal{N}_t, p_t^D, p_t^C), \tag{8}$$

where $\mathcal{N}_t$ denotes the search needs inferred at step t, p_t^D denotes the compact-memory retrieval sub-plan, and p_t^C denotes an optional source-access sub-plan. The search needs specify what evidence should be found and are updated as the evidence state changes across retrieval steps.

For compact-memory retrieval, the compact-memory sub-plan is defined as

$$p_t^D = (\mathcal{Q}_t^{\text{lex}}, \mathcal{Q}_t^{\text{sem}}), \tag{9}$$

where $\mathcal{Q}_t^{\text{lex}}$ contains lexical queries and $\mathcal{Q}_t^{\text{sem}}$ contains semantic queries. These queries are constructed according to the search needs $\mathcal{N}_t$ and provide retrieval-ready inputs for sparse and dense retrieval, respectively. Guided by p_t^D, GraP-Mem retrieves compact semantic memory documents using both the original question and the plan-derived retrieval materials. The retrieved results are deduplicated and fused into a compact-memory candidate set $\mathcal{B}_t^D$, which provides the first layer of evidence for the current question.

When the evidence gap g_t or the inferred search needs $\mathcal{N}_t$ indicate that compact memories are insufficient, the Plan Agent activates the source-access sub-plan:

$$p_t^C = (\mathcal{O}_t, \mathcal{Q}_t^C), \tag{10}$$

where $\mathcal{O}_t$ specifies which linked source contexts should be opened, and $\mathcal{Q}_t^C$ denotes source-level queries for retrieving additional source contexts. Opening linked contexts recovers details from already localized sessions, whereas source-level retrieval searches the source collection when the missing evidence cannot be resolved from retrieved compact memories. Guided by p_t^C, GraP-Mem obtains a source-context candidate set $\mathcal{B}_t^C$, which provides fine-grained evidence such as exact wording, numerical values, temporal order, and local event dependencies. If source access is not needed, $p_t^C = \emptyset$.

The compact-memory candidates $\mathcal{B}_t^D$ and the optional source-context candidates $\mathcal{B}_t^C$ are then passed to the Integration Agent. In this way, the Plan Agent controls both the retrieval objective and the memory granularity to access, allowing GraP-Mem to reduce unnecessary context exposure while preserving access to detailed source evidence when needed.

4.4 Evidence Memory Integration

Plan-guided memory access produces candidates from different memory granularities, but directly appending all retrieved content to the prompt would introduce redundancy and noise. GraP-Mem therefore maintains an evidence working memory, denoted by the question-specific evidence state $\mathcal{E}_t$. This evidence state stores useful information collected so far and serves as the shared context for subsequent planning and final answer generation.

At retrieval step t, after the plan p_t is executed, the Integration Agent receives the current evidence state $\mathcal{E}_t$ and the newly retrieved candidates, including compact-memory candidates $\mathcal{B}_t^D$ and optional source-context candidates $\mathcal{B}_t^C$. It updates the evidence state and produces a sufficiency decision:

$$(\mathcal{E}_{t+1}, z_{t+1}, g_{t+1}) = \Phi_{\text{int}}(q, \mathcal{E}_t, \mathcal{B}_t^D, \mathcal{B}_t^C), \tag{11}$$

where z_{t+1} indicates whether the updated evidence is sufficient for answering the question, and g_{t+1} denotes the remaining evidence gap when the evidence is still incomplete. If no source context is accessed at step t, $\mathcal{B}_t^C = \emptyset$.

The integration update is question-conditioned. Instead of summarizing all retrieved candidates uniformly, the Integration Agent extracts answer-relevant evidence, filters redundant or irrelevant information, and preserves details needed for reliable generation, such as temporal relations, numerical values, decisions, and answer-bearing facts. In this way, $\mathcal{E}_{t+1}$ becomes a focused intermediate representation that abstracts useful information from retrieved memories while avoiding unnecessary context exposure.

The sufficiency decision closes the loop between retrieval planning and evidence integration. If $z_{t+1} = 1$, GraP-Mem stops retrieval and uses $\mathcal{E}_{t+1}$ as the final evidence state. Otherwise, the evidence gap g_{t+1} is returned to the Plan Agent, which uses it together with the updated evidence state and access history to guide the next memory-access step. The access history is updated to $\mathcal{H}_{t+1}$ with the access decisions executed at the current step. If the retrieval budget is reached before the evidence is judged sufficient, GraP-Mem uses the latest evidence state for final generation. Thus, each round refines the existing evidence state rather than restarting retrieval from scratch, allowing GraP-Mem to accumulate complementary evidence across compact memories and source contexts.

Finally, the Answer Model generates the output

Model	Method	Single-hop		Multi-hop		Open-domain		Temporal		Average	
		F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU
GPT-4o-mini	Long-LLM	46.68	37.54	29.23	22.76	16.87	13.70	25.97	19.42	37.31	29.57
	RAG	52.19	46.80	32.17	23.59	23.21	18.88	30.77	25.99	42.25	36.47
	Mem0	47.65	38.72	38.72	27.13	28.64	21.58	48.93	40.51	45.10	35.90
	MemoryOS	48.62	42.99	35.27	25.22	20.02	15.52	41.15	30.76	42.84	35.47
	LightMem	41.79	37.83	29.78	24.80	16.89	13.92	43.71	39.72	38.44	34.35
	A-Mem	44.65	37.06	27.02	20.09	12.14	12.00	45.85	36.67	39.65	32.31
	HippoRAG	54.84	48.84	33.59	25.46	28.59	**23.89**	48.17	39.32	47.92	41.02
	CAM	50.58	44.36	33.55	24.18	18.23	12.77	44.14	38.28	44.10	37.43
	GraP-Mem	**59.98**	**54.35**	**46.18**	**39.02**	**28.65**	21.93	**61.72**	**54.81**	**55.86**	**48.53**
Qwen2.5-14B	Long-LLM	46.05	39.56	32.08	24.46	14.89	11.41	30.51	24.45	38.31	31.89
	RAG	49.79	43.95	28.11	21.43	20.42	17.40	24.73	20.02	38.77	33.18
	Mem0	42.58	35.15	31.73	24.82	15.03	11.28	28.96	26.24	36.04	29.91
	MemoryOS	46.33	41.62	38.19	29.26	20.27	15.94	32.24	27.86	40.28	34.89
	LightMem	34.92	31.22	25.45	19.61	15.81	11.81	32.03	27.70	31.39	27.15
	A-Mem	33.75	30.04	22.09	15.28	13.49	10.74	27.19	22.05	28.98	24.47
	HippoRAG	42.45	37.14	27.57	20.62	19.74	15.81	30.66	26.33	35.85	30.53
	CAM	50.39	45.59	34.50	24.62	23.86	20.84	44.70	36.30	44.64	38.27
	GraP-Mem	**59.85**	**52.39**	**55.05**	**48.12**	**33.58**	**28.67**	**48.32**	**41.85**	**53.48**	**46.65**

Table 1: Performance comparison on the LoCoMo benchmark, covering single-hop, multi-hop, open-domain, and temporal settings. F1 and BLEU-1 scores (%) are reported. Best results are highlighted in bold, and second-best results are underlined.

Model	Method	F1	BLEU
GPT-4o-mini	RAG	25.32	21.23
	Mem0	29.12	22.84
	MemoryOS	25.12	21.08
	A-Mem	27.04	22.46
	LightMem	17.23	15.23
	HippoRAG	28.24	22.85
	CAM	33.15	29.20
	GraP-Mem	**39.42**	**35.47**
Qwen2.5-14B	RAG	20.29	17.04
	Mem0	26.38	21.87
	MemoryOS	23.22	19.86
	A-Mem	25.42	21.05
	LightMem	16.87	12.42
	HippoRAG	22.56	18.53
	CAM	28.18	23.81
	GraP-Mem	**35.42**	**29.84**

Table 2: Results on 300 questions belonging to 10 documents randomly sampled from NarrativeQA. The subset is used because the full test set contains over 10,000 questions, making full long-context evaluation prohibitively expensive.

only from the query and the final evidence state:

$$y = F(q, \mathcal{E}_T). \tag{12}$$

The final generation step is therefore conditioned on integrated evidence rather than all retrieved memories or complete source contexts. This design preserves progressively collected evidence while reducing redundant context and improving the reliability of memory-grounded answering.

5 Experiment

5.1 Experimental Settings

Evaluation Benchmarks We evaluate GraP-Mem on two long-context reasoning benchmarks, LoCoMo and NarrativeQA. LoCoMo evaluates long-term conversational question answering, where models need to track information across extended multi-turn interactions. NarrativeQA focuses on narrative understanding and requires models to retrieve and reason over evidence from long documents. These two benchmarks cover both dialogue-based memory reasoning and document-level long-context reasoning, providing a comprehensive testbed for evaluating memory systems. More details about the datasets are provided in Appendix A.1.

Baselines. We compare GraP-Mem with representative memory-augmented agent methods. The baselines include non-graph memory systems: Long-LLM, RAG, Mem0 (Chhikara et al., 2025), MemoryOS (Kang et al., 2025) and LightMem(Fang et al., 2025b), as well as graph-based or structured memory methods, including HippoRAG (Gutiérrez et al., 2025), A-Mem (Xu et al.,

Configuration	Single Hop		Multi-hop		Open Domain		Temporal	
	F1	Diff	F1	Diff	F1	Diff	F1	Diff
Full GraP-Mem (Ours)	**59.98**	-	**46.18**	-	**28.65**	-	**61.72**	-
w/o Compact Semantic Memory	57.25	(↓4.6%)	42.01	(↓9.0%)	24.26	(↓15.3%)	54.47	(↓11.7%)
w/o Source-Context Expansion	54.58	(↓9.0%)	42.19	(↓8.6%)	25.07	(↓12.5%)	54.78	(↓11.2%)
w/o Planning	57.98	(↓3.3%)	40.48	(↓12.3%)	17.46	(↓39.1%)	50.18	(↓18.7%)
w/o Evidence Integration	58.21	(↓3.0%)	42.06	(↓8.9%)	23.72	(↓17.2%)	52.36	(↓15.2%)

Table 3: Ablation study of different components in our GraP-Mem architecture. Performance drops are highlighted.

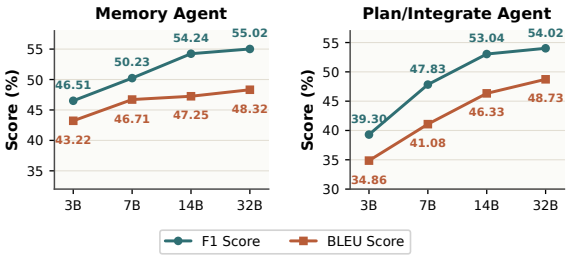

Figure 2: Effect of agent backbone size on Memory Agent and Plan/Integrate Agent.

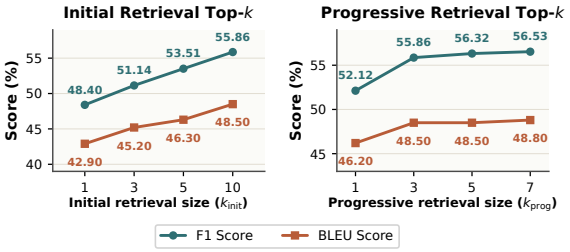

Figure 3: Effect of initial and progressive retrieval top-k on GraP-Mem.

2025), and CAM (Li et al., 2025a). For each baseline, we use the official implementation when available. Otherwise, we follow the reported experimental settings as closely as possible. Additional implementation details are provided in Appendix A.2.

Implementation Details. We instantiate GraP-Mem and all baselines with both closed-source and open-source backbone models. Specifically, we use GPT-4o-mini as the closed-source model and Qwen2.5-14B-Instruct as the open-source model. GPT-4o-mini is accessed through the official API, while Qwen2.5-14B-Instruct is deployed locally with vLLM. For embedding-based retrieval, all methods use BGE-M3 to ensure a consistent retrieval backbone across different systems.

5.2 Main Results

Table 1 reports the results on LoCoMo across four question types and two backbone models. GraP-Mem achieves the highest average performance in both settings. With GPT-4o-mini, GraP-Mem obtains 55.86 Average F1 and 48.53 Average BLEU, improving over HippoRAG by 7.94 F1 and 7.51 BLEU points. With Qwen2.5-14B, GraP-Mem reaches 53.48 Average F1 and 46.65 Average BLEU, exceeding CAM by 8.84 F1 and 8.38 BLEU points. These results show that GraP-Mem brings consistent gains across different backbone models.

Across question types, GraP-Mem achieves the highest F1 in all categories on both backbone models. The gains are especially clear on reasoning-intensive questions. With GPT-4o-mini, GraP-Mem improves Multi-hop F1 from 38.72 to 46.18 and Temporal F1 from 48.93 to 61.72 compared with Mem0. With Qwen2.5-14B, GraP-Mem raises Multi-hop F1 from 38.19 to 55.05 compared with MemoryOS and Open-domain F1 from 23.86 to 33.58 compared with CAM. These results indicate that GraP-Mem is effective at combining dispersed evidence and recovering fine-grained information from long histories.

Table 2 further evaluates GraP-Mem on NarrativeQA, which requires long-range narrative understanding and evidence aggregation. GraP-Mem consistently achieves the best results under both backbone models. With GPT-4o-mini, it obtains 39.42 F1 and 35.47 BLEU, outperforming CAM by 6.27 F1 and 6.27 BLEU points. With Qwen2.5-14B, it reaches 35.42 F1 and 29.84 BLEU, exceeding CAM by 7.24 F1 and 6.03 BLEU points. These results suggest that GraP-Mem generalizes beyond dialogue-based memory tasks and remains effective for long-document reasoning.

Overall, the results support the design of plan-

guided multi-granularity memory access. GraP-Mem first uses compact semantic memory for efficient localization, selectively expands to source contexts when more evidence is needed, and maintains a question-specific evidence state for final answer generation.

5.3 Ablation Study

We conduct a module-level ablation study on LoCoMo to evaluate the contribution of each component in GraP-Mem. As shown in Table 3, removing any module consistently reduces F1 across all question types, indicating that GraP-Mem benefits from the joint design of compact semantic memory, source-context expansion, planning, and evidence integration.

Compact semantic memory mainly supports efficient session localization. Removing it causes clear drops on open-domain and temporal questions, with F1 decreasing by 15.3% and 11.7%, respectively. Source-context expansion is also important for evidence completeness: disabling it reduces F1 by 12.5% on open-domain questions and 11.2% on temporal questions, showing that compact memories alone cannot always recover fine-grained evidence.

Planning and evidence integration are especially important for complex queries. Removing planning leads to the largest degradation on open-domain questions, where F1 drops by 39.1%, and also substantially hurts temporal reasoning with an 18.7% drop. Removing evidence integration decreases open-domain and temporal F1 by 17.2% and 15.2%, respectively, confirming the importance of maintaining a focused evidence state. Overall, these results show that GraP-Mem improves performance by combining retrieval-friendly compact memories, selective source-context recovery, query-aware planning, and evidence-centered integration.

5.4 Further Analysis

Impact of Model Size. We further analyze how the backbone size of different agents affects GraP-Mem. As shown in Figure 2, increasing the model size improves both F1 and BLEU for the Memory Agent and the Plan/Integrate Agent, indicating that GraP-Mem can benefit from stronger LLM backbones. Scaling the Memory Agent from 3B to 32B raises F1 from 46.51 to 55.02, suggesting that larger models construct higher-quality compact semantic memories by better identifying salient information and rewriting context-dependent content into self-contained records. The Plan/Integrate Agent is more sensitive to model scale, with F1 increasing from 39.30 to 54.02. This larger gain is consistent with its inference-time role, which requires identifying evidence gaps, scheduling memory access across different granularities, and updating the evidence working memory. Overall, the results show that model capacity is especially important for planning and evidence integration, while memory construction remains relatively stable with smaller models.

Impact of Hyperparameter. Figure 3 studies the sensitivity of GraP-Mem to the retrieval size in the initial and progressive retrieval stages. Increasing k_{init} consistently improves performance, raising F1 from 48.40 to 55.86, which shows that broader initial recall helps locate relevant sessions and construct stronger initial evidence. Progressive retrieval shows a different trend: performance improves clearly from $k_{\text{prog}} = 1$ to 3, but becomes nearly saturated when k_{prog} further increases. This indicates that, after the evidence gap is identified, GraP-Mem can recover missing details with a small number of focused candidates. These results support the effectiveness of combining sufficient initial recall with gap-aware progressive memory access.

6 Conclusion

We present GraP-Mem, a multi-granularity memory planning framework for long-horizon LLM agents. GraP-Mem reframes memory access as a planned evidence construction process rather than static retrieval. By separating memory preservation from memory exposure, it stores compact semantic memory documents as the default retrieval view while retaining source contexts for detailed evidence recovery. During inference, a Plan Agent adaptively schedules memory access according to query-specific evidence needs, and an Integration Agent maintains a focused evidence state for final answer generation. Experiments on LoCoMo and NarrativeQA demonstrate consistent improvements across backbone models and show the effectiveness of GraP-Mem in both dialogue-based memory and long-document reasoning settings. These results highlight the importance of adaptive memory exposure for building reliable and efficient long-horizon LLM agents.

Limitations

Long-horizon memory agents still face several open challenges. Memory construction depends on reliable identification and preservation of useful historical information, and although GraP-Mem combines compact semantic memories with source contexts, more robust memory writing may improve stability across domains. Adaptive memory access also introduces a trade-off between evidence completeness and inference efficiency when complex queries require iterative retrieval. In addition, current evaluation mainly covers representative dialogue and long-document benchmarks, and broader validation in interactive and real-world agent environments remains future work.

Ethical Considerations

This work studies long-horizon memory mechanisms for LLM agents and uses only publicly available benchmarks, LoCoMo and NarrativeQA. We do not introduce new datasets or intentionally collect sensitive personal information. In real-world deployment, memory agents may store user-related histories, so GraP-Mem should be used with privacy protection, data filtering, and user control over retained memories.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pages 9112–9141.

Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. 2025. Reinforcement learning for long-horizon interactive llm agents. *arXiv preprint arXiv:2502.01600*.

X Chen, Jiakai TANG, Xu CHEN, and Yankai LIN. 2024. A survey on large language model based autonomous agents. In *CCL 2024–23rd Chinese natl conf comput linguist*, volume 2, pages 141–150.

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*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Jinyuan Fang, Yanwen Peng, Xi Zhang, Yingxu Wang, Xinhao Yi, Guibin Zhang, Yi Xu, Bin Wu, Siwei Liu, Zihao Li, and 1 others. 2025a. A comprehensive survey of self-evolving ai agents: A new paradigm bridging foundation models and lifelong agentic systems. *arXiv preprint arXiv:2508.07407*.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, and 1 others. 2025b. Lightmem: Lightweight and efficient memory-augmented generation. *arXiv preprint arXiv:2510.18866*.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tian Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2(3).

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*.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, and 1 others. 2024. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, and 1 others. 2025. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 7969–7992.

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 25972–25981.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Rui Li, Zeyu Zhang, Xiaohe Bo, Zihang Tian, Xu Chen, Quanyu Dai, Zhenhua Dong, and Ruiming Tang. 2025a. Cam: A constructivist view of agentic memory for llm-based reading comprehension. *arXiv preprint arXiv:2510.05520*.

Zhiyu Li, Chenyang Xi, Chunyu Li, Ding Chen, Boyu Chen, Shichao Song, Simin Niu, Hanyu Wang, Jiawei Yang, Chen Tang, and 1 others. 2025b. Memos: A memory os for ai system. *arXiv 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.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv:2402.17753*.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T Le, Samira Daruki, Xiangru Tang, and 1 others. 2025. Reasoningbank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*.

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

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, and 1 others. 2025. Agent kb: Leveraging cross-domain experience for agentic problem solving. *arXiv preprint arXiv:2507.06229*.

Bing Wang, Xinnian Liang, Jian Yang, Hui Huang, Zhenhe Wu, ShuangZhi Wu, Zejun Ma, and Zhoujun Li. 2025a. Scm: Enhancing large language model with self-controlled memory framework. In *International Conference on Database Systems for Advanced Applications*, pages 188–203. Springer.

Yu Wang, Chi Han, Tongtong Wu, Xiaoxin He, Wangchunshu Zhou, Nafis Sadeq, Xiusi Chen, Zexue He, Wei Wang, Gholamreza Haffari, and 1 others. 2024. Towards lifespan cognitive systems. *arXiv preprint arXiv:2409.13265*.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. 2025b. Mem-{\alpha}: Learning memory construction via reinforcement learning. *arXiv preprint arXiv:2509.25911*.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, and 1 others. 2025c. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*.

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

BY Yan, Chaofan Li, Hongjin Qian, Shuqi Lu, and Zheng Liu. 2025. General agentic memory via deep research. *arXiv preprint arXiv:2511.18423*.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):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 Experiment detail

A.1 Dataset Descriptions

LoCoMo LoCoMo(Maharana et al., 2024) is a long-term conversational benchmark designed to evaluate memory access, information integration, and reasoning over extended agent interactions. It contains 10 multi-session dialogues, where each dialogue spans many sessions and hundreds of turns, with an average length of approximately 600 turns and 16K tokens. The QA split provides questions with annotated evidence locations and covers several evaluation categories, including single-hop reasoning, multi-hop reasoning, temporal reasoning, and open-domain knowledge reasoning. These categories test whether a model can retrieve local evidence, connect information across distant turns, reason over event order, and combine dialogue context with external knowledge. Following common experimental practice in prior LoCoMo evaluations, we exclude adversarial questions from our experiments and report results on the remaining QA categories(Chhikara et al., 2025).

NarrativeQA NarrativeQA(Kočiskỳ et al., 2018) is a long-context reading comprehension benchmark designed to evaluate the ability of models to understand and reason over extended narrative documents, including books and movie scripts. The full dataset contains more than one thousand story documents and tens of thousands of human-written question–answer pairs. Unlike tasks that can be solved through local lexical matching, NarrativeQA requires models to integrate information across the global structure of a narrative and infer answers from dispersed evidence. The questions are derived from human-written abstractive summaries, which further encourages high-level narrative understanding rather than reliance on surface-level context overlap. In our experiments, we randomly sample 10 long documents from the NarrativeQA corpus and evaluate the associated 300 QA pairs. This subset-based evaluation follows prior practice and is necessary because the full test set contains 10,557 questions, making exhaustive evaluation computationally expensive. The sampled documents contain approximately 60,000 tokens on average, providing a challenging setting for assessing long-range comprehension and coherent evidence aggregation.

A.2 Baseline

RAG: A basic retrieval baseline that treats the complete input context as external memory. It encodes the context with the BGE-M3 embedding model and stores the resulting vectors for later retrieval, without introducing additional memory compression, update, or structural organization mechanisms.

Mem0(Chhikara et al., 2025): A scalable long-term memory system designed to support continuous information accumulation from ongoing dialogues and interaction streams. It extracts salient facts from new contexts, stores them as compact memory entries, and updates existing entries when related information is observed. By merging similar facts and removing redundant content, Mem0 maintains a concise memory store and retrieves only information that is relevant to the current query instead of repeatedly processing the full context. It also supports an optional graph memory extension for modeling relations among memory facts.

MemoryOS(Kang et al., 2025): A hierarchical memory architecture developed for AI agents that operate over long conversational histories. It separates memory into short-, mid-, and long-term stores, allowing recent context, intermediate summaries, and persistent user-related information to be managed at different levels. The system coordinates memory storage, dynamic update, adaptive retrieval, and response generation, with the goal of maintaining contextual continuity, coherent interaction history, and personalized responses across extended dialogues.

HippoRAG(Gutiérrez et al., 2025): A graph-based retrieval-augmented generation framework motivated by hippocampal indexing theory. It first converts documents into a knowledge graph, where concepts and relations provide a structured representation of the input corpus. During retrieval, HippoRAG uses Personalized PageRank over concept seeds to propagate relevance signals across the graph, enabling information from distant but related contexts to be connected in a single retrieval step. This design is intended to improve multi-hop retrieval and reasoning over dispersed evidence compared with standard vector-based RAG.

A-Mem(Xu et al., 2025): An agentic memory system that organizes user–agent interactions into an evolving memory network. Inspired by Zettelkasten-style note organization, it converts

newly observed memories into structured notes with multiple attributes and links them to related historical entries. These associations are established through embedding similarity analysis and LLM-based reasoning, allowing the memory structure to evolve continuously as new experiences are added. This design supports contextual organization beyond fixed memory schemas.

LightMem(Fang et al., 2025b): A lightweight memory system inspired by human memory processes. It introduces a three-stage memory mechanism that filters sensory information, organizes topic-aware short-term memory, and updates long-term memory during offline consolidation. By reducing unnecessary memory operations while preserving useful information, LightMem aims to improve memory processing efficiency without substantially sacrificing task performance.

CAM(Li et al., 2025a): A structured memory framework based on constructivist theory. It organizes memory through hierarchical summaries and overlapping clusters, allowing related information to be grouped while still preserving flexible connections across memory units. During retrieval, CAM explores the memory structure in an associative manner, which helps integrate relevant evidence from long contexts and improves efficiency for long-text reading tasks.

B Use of AI Assistants

We use ChatGPT to improve the presentations of this paper