RAMEM: CONTEXTUAL REINSTATEMENT FOR LONG-TERM AGENTIC MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory has become increasingly important for LLM agents that operate across extended interactions and evolving task contexts. Recent memory systems have made past experiences more persistent, compact, and retrievable, yet memory availability does not guarantee that the intended past episode remains identifiable. When semantically similar experiences from different episodes are compressed into reusable fragments, content-centric recall can marginalize the contextual distinctions that separate them. We refer to this failure as **context collapse**: episodically distinct memories become aliased under content-based retrieval, allowing contextually incompatible memories to compete with the intended trace. To address this problem, we propose **Contextual Reinstatement for Agentic Memory (RaMem)**, which restores long-term recall in a joint content–context space. RaMem represents each memory as content bound to the episodic coordinates under which it was observed, and uses cues in the current query to reinstate the contextual coordinates needed to disambiguate recall. Because such cues may be incomplete or uncertain, **Conservative Contextual Reinstatement** activates contextual constraints only when they can be reliably grounded. RaMem then performs **Validity-Constrained Recall**, conditioning semantic relevance on compatibility with the reinstated context while retaining content-relevant memories as fallback, and preserves the resulting content–context binding during downstream reasoning. Experiments on long-term memory benchmarks show that RaMem consistently outperforms strong memory baselines, with average F1 gains exceeding 10% across several backbones. Code is available at `https://anonymous.4open.science/r/ramem_repo-F4C4/`.

1 INTRODUCTION

Long-term memory has become a central component of agentic systems, enabling agents to carry useful experience across interactions and reason beyond a single context window. Existing work has advanced agent memory along several complementary directions. Some systems maintain persistent natural-language records, virtual contexts, or user-level facts across sessions (Park et al., 2023; Packer et al., 2024; Zhong et al., 2024; Chhikara et al., 2025). Others introduce more structured organizations, such as temporal graphs, linked notes, symbolic representations, or multi-view indexes, to support memory consolidation and access (Rasmussen et al., 2025; Liu et al., 2026). Recent agentic memory systems study how memory should be actively managed during task execution, including when to distill experience, reuse procedural knowledge, retrieve prior information, or update stored memories (Ma et al., 2025; Cao et al., 2025; Yu et al., 2026; Du et al., 2025a). Across these directions, a common objective is to make past experience accessible when useful again. At a high level, this has framed long-term memory primarily as an *availability problem*: what information should be retained, how it should be organized, and how it can be surfaced when needed.

Yet memory availability does not guarantee that the correct past episode remains identifiable. Most memory systems implicitly rely on a *content-sufficiency assumption*: once an experience is compressed into a self-contained memory fragment, matching its content to a query is sufficient to recover the appropriate occurrence (Chhikara et al., 2025; Liu et al., 2026). Long-term experience, however, is not a collection of context-free fragments (Tulving, 2002). The same person, preference, plan, activity, or event type may recur across different sessions and interaction states, producing memories that are semantically similar but episodically distinct. Their identity therefore depends not only on *what* was

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

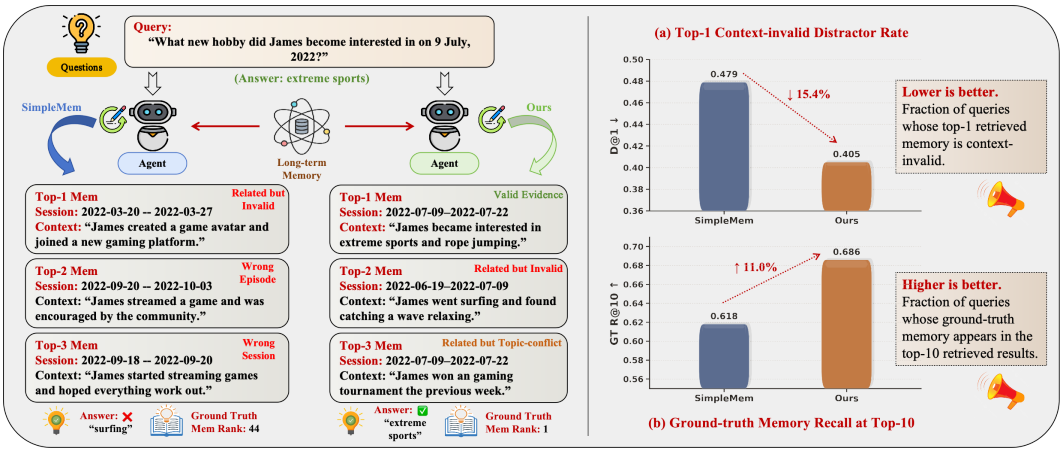


Figure 1: Context collapse in long-term agent memory. **Left**: a representative case where content-centric recall retrieves semantically related memories from incorrect episodes, causing the target episode to be obscured by contextually incompatible alternatives. **Right**: aggregate results on the context-confusable subset show that this failure is systematic: SimpleMem frequently ranks wrong-episode memories first and places the ground-truth memory substantially lower.

remembered, but also on the context in which it was observed. For example, when it occurred or was mentioned, who was involved, and which interaction episode it belonged to. When recall operates primarily over remembered content, these contextual distinctions can be marginalized away (Liu et al., 2025). We call this failure mode **context collapse**: episodically distinct memories become aliased under content-centric recall, allowing semantically relevant but contextually incompatible memories to compete with the intended trace. Figure 1 illustrates this, where memories from incorrect sessions remain highly relevant in content and can dominate recall over the target episode.

Context collapse exposes a limitation that cannot always be repaired by improving semantic relevance alone, because the information needed to distinguish competing memories may lie in their contextual coordinates rather than content. This observation is closely aligned with classical accounts of episodic memory and encoding specificity, which emphasize that successful recall depends on the correspondence between a retrieval cue and the context in which an experience was encoded (Tulving, 2002; Tulving & Thomson, 1973). Work on temporal context and event segmentation similarly highlights contextual states and event boundaries as organizing principles of memory (Howard & Kahana, 2002; Kurby & Zacks, 2008). These perspectives suggest a different computational principle for long-term agent memory: rather than treating stored memories as context-free fragments and relying on relevance alone, recall should *reinstate* the query-relevant contextual conditions that distinguish the intended episode. We refer to this principle as **contextual reinstatement**.

Motivated by this principle, we propose **Contextual Reinstatement for Agentic Memory (RaMem)**, which restores long-term recall from a content-centric projection to a joint content–context space. RaMem represents each memory as content bound to its episodic coordinates and uses query cues to partially reinstate the context needed to identify the intended episode during long-horizon interactions. Because such cues may be incomplete or ambiguous, **Conservative Contextual Reinstatement** allows context to constrain recall only when it can be reliably grounded. RaMem then performs **Validity-Constrained Recall**, conditioning semantic relevance on compatibility between the reinstated query context and stored episodic coordinates while retaining content-relevant candidates as fallback. The recalled memories preserve their content–context binding during downstream reasoning, preventing the recovered episodic identity from being lost again after recall.

The main contributions of this work are as follows:

- We identify **Context Collapse** as a fundamental failure mode of long-term agent memory, where episodically distinct memories become aliased under content-centric recall as their contextual distinctions are lost.

- We propose **Contextual Reinstatement for Agentic Memory (RaMem)**, which recalls memories in a joint content–context space through Conservative Contextual Reinstatement, Validity-Constrained Recall, and preservation of content–context binding.
- Experiments on long-term memory benchmarks show that RaMem consistently improves answer quality, memory retrieval, and context efficiency, while substantially reducing context-invalid distractors.

2 Related Work

Long-term memory has become an important mechanism for extending LLM agents beyond a single interaction. Early systems maintain external memory streams, virtual context, or persistent user facts, enabling agents to carry information across sessions (Park et al., 2023; Shinn et al., 2023; Packer et al., 2024; Zhong et al., 2024). Later work makes these memories more structured by using temporal knowledge graphs, note-like organizations, graph-augmented facts, lightweight memory modules, and multi-view indexes (Rasmussen et al., 2025; Xu et al., 2025; Chhikara et al., 2025; Fang et al., 2025; Liu et al., 2026). Recent methods further study how agents should distill experience, reuse procedural knowledge, learn memory operations, or control retrieval and reflection during task execution (Nan et al., 2025; Yu et al., 2026; 2025; Cao et al., 2025; Du et al., 2025a). These directions have made agent memory more persistent, compact, and easier to retrieve. In parallel, long-context modeling and retrieval-augmented generation improve how models access large information sources through context extension, compression, retrieval, graph-based evidence aggregation, and agentic retrieval policies (Lewis et al., 2020; Liu et al., 2024; Jiang et al., 2023a; Asai et al., 2023; Edge et al., 2024; Guo et al., 2024; Yao et al., 2022). A more detailed discussion is provided in Appendix A.

3 Method

In this section, we present RaMem in detail, with an overview illustrated in Figure 2. The core idea is to restore query-relevant episodic context before recall and preserve the resulting content–context binding during downstream reasoning.

3.1 Context-Bound Memory Space and Context Collapse

Consider a long-term memory bank $\mathcal{M} = \{m_i\}_{i=1}^{N}$. Rather than treating each memory as a decontextualized fragment, we represent it as a context-bound episodic trace $m_i = (x_i, e_i) \in \mathcal{X} \times \mathcal{E}$, where x_i is the remembered content and e_i its episodic coordinates. In our instantiation,

$$e_i = \{\tau_i^{\text{event}}, \tau_i^{\text{mention}}, \tau_i^{\text{session}}, p_i, \ell_i, u_i, z_i\},$$

covering event time, mentioned time, session span, participants, location, entities, and topic. These coordinates are part of the memory's episodic identity rather than auxiliary metadata, as they specify the context needed to distinguish otherwise similar experiences.

This distinction becomes critical when similar content recurs across episodes. Two memories $m_i = (x_i, e_i)$ and $m_j = (x_j, e_j)$ may be semantically close in $\mathcal{X}$ despite originating from different contexts in $\mathcal{E}$. Conventional relevance-based recall primarily observes the content projection $\pi_{\mathcal{X}} : \mathcal{X} \times \mathcal{E} \to \mathcal{X}$ with $\pi_{\mathcal{X}}(m_i) = x_i$, causing episodic distinctions to be marginalized during retrieval.

We refer to the resulting loss of episodic identifiability as **context collapse**. Let $A_q(e_i) \in \{0, 1\}$ indicate whether memory m_i is contextually admissible for query q. A context-collapse pair occurs when

$$d_{\mathcal{X}}(x_i, x_j) \le \epsilon, \qquad A_q(e_i) \neq A_q(e_j), \tag{1}$$

meaning that two memories are nearly indistinguishable under content-centric recall despite differing in query-relevant episodic context. Here $d_{\mathcal{X}}$ denotes the similarity induced by the underlying retriever; neither $\mathcal{X}$ nor $\mathcal{E}$ is assumed to follow a particular geometric parameterization.

Context collapse therefore cannot always be resolved by stronger semantic matching: the information needed to distinguish competing memories may lie in e_i rather than x_i. RaMem addresses this mismatch by using query cues to partially recover the contextual coordinates discarded by $\pi_{\mathcal{X}}$. We next introduce *Conservative Contextual Reinstatement*, which performs this recovery only when those coordinates can be grounded reliably.

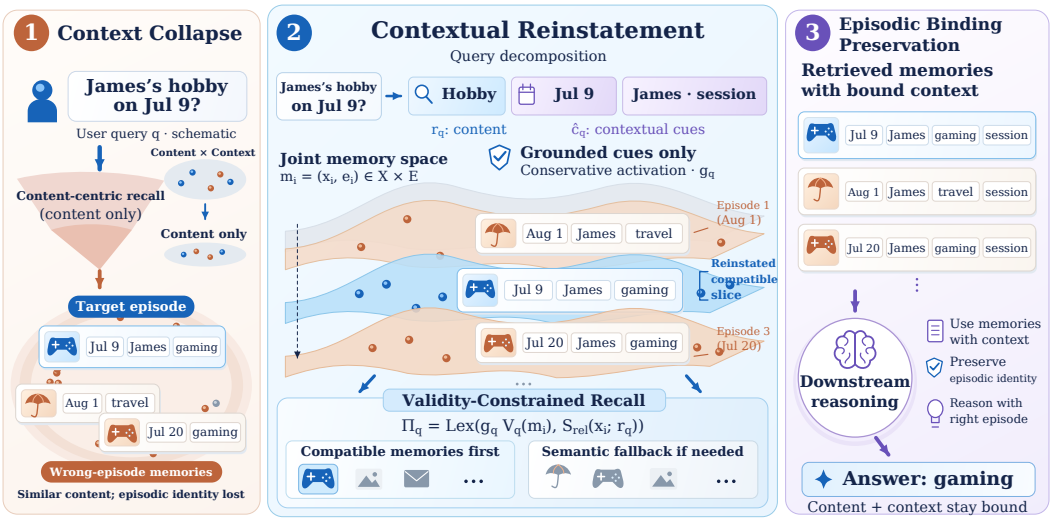


Figure 2: **Overview of RaMem**. **(1) Context Collapse**: episodically distinct memories become aliased under content-centric recall when contextual distinctions are projected away. **(2) Contextual Reinstatement**: RaMem reconstructs query-relevant contextual coordinates, reinstates the compatible episodic slice in joint content–context space, and performs validity-constrained recall with semantic fallback. **(3) Episodic Binding Preservation**: recalled content remains bound to episodic coordinates during downstream reasoning, preserving the contextual identity required for correct recall.

3.2 Conservative Contextual Reinstatement

Context collapse arises because conventional recall observes only a projection of a context-bound memory trace. Recovering the missing episodic coordinates therefore requires information beyond memory content; the current query provides cues about *which part of the past* should be reinstated. Given a query q, RaMem applies

$$\mathcal{R}(q) = (r_q, \hat{c}_q, g_q)\,, \tag{2}$$

where r_q specifies *what* should be recalled, $\hat{c}_q$ reconstructs *from which episode*, and $g_q \in \{0, 1\}$ indicates whether the reconstructed context is sufficiently grounded to constrain recall. We represent $\hat{c}_q = \{\hat{\tau}_q, \hat{p}_q, \hat{\ell}_q, \hat{u}_q, \hat{z}_q\}$, covering grounded temporal, participant, location, entity, and episode or topic cues when available.

Geometrically, $\mathcal{R}$ performs a query-conditioned *lifting* from the projected content space back toward the joint content–context space: $q \mapsto (r_q, \hat{c}_q)$ with $\hat{c}_q \subseteq \mathcal{E}$. The lifting is deliberately partial: RaMem restores only the contextual coordinates needed to distinguish memories that would otherwise remain aliased under content-centric recall.

Because contextual cues may be ambiguous or underspecified, imposing every inferred coordinate as a hard constraint can suppress the correct episode. We therefore make reinstatement *conservative*:

$$g_q = \mathbb{I}\left[\mathrm{Ground}(q, \hat{c}_q)\right], \tag{3}$$

where $\mathrm{Ground}(\cdot)$ determines whether the required contextual coordinates are supported by observable query cues and the interaction history. When $g_q = 1$, the reinstated context may constrain recall; otherwise, RaMem falls back to content-based recall. This selective activation is crucial because correct context can disambiguate nearly identical memories, whereas an incorrect constraint may exclude the target episode entirely.

In our implementation, session-level temporal context provides the primary explicit reinstatement coordinate. Temporal cues are grounded to a range $\hat{\tau}_q$, while participant, entity, location, and topic information provide auxiliary coordinates. Temporal reinstatement is activated only when a temporal signal is present and can be resolved reliably. Once context is reinstated, RaMem uses it to constrain semantic recall, which we formalize next as *Validity-Constrained Recall*.

3.3 VALIDITY-CONSTRAINED RECALL

Conservative reinstatement recovers contextual coordinates that distinguish memories otherwise confounded in content space. RAMEM uses these coordinates to separate *semantic relevance* from *contextual validity*. Given the reinstated context $\hat{c}_q$, we define

$$V_q(m_i) = C(e_i, \hat{c}_q) \in \{0, 1\}, \tag{4}$$

where $C(\cdot, \cdot)$ measures whether the episodic coordinates of memory $m_i = (x_i, e_i)$ are compatible with the context implied by the query. The memories satisfying $V_q(m_i) = 1$ form the query-specific contextual region in which recall should preferentially operate.

This yields the central recall principle of RAMEM:

$$\boxed{m_q^\star = \arg\max_{m_i \in \mathcal{M}} S_{\text{rel}}(x_i; r_q) \quad \text{s.t.} \quad V_q(m_i) = 1} \tag{5}$$

whenever contextual reinstatement is reliably grounded. Here relevance determines which content addresses the query, whereas validity determines whether that content belongs to a compatible episodic context. Thus, RAMEM does not replace semantic relevance with contextual matching; it conditions semantic recall on episodic compatibility.

Because $\hat{c}_q$ may be partial or noisy, we realize Eq. (5) conservatively as a priority relation rather than irreversible exclusion:

$$\Pi_q(m_i) = \text{Lex}\left(g_q V_q(m_i), S_{\text{rel}}(x_i; r_q)\right). \tag{6}$$

When $g_q = 1$, context-compatible memories are prioritized and semantic relevance resolves the remaining ordering; when $g_q = 0$, recall naturally reduces to relevance-based ranking. This preserves content-relevant memories as fallback while allowing grounded contextual information to disambiguate competing episodes.

In our implementation, S_{rel} is instantiated with complementary semantic and lexical retrieval signals, while session-level time provides the primary explicit validity coordinate. Temporal compatibility is determined from overlap between the grounded query interval and each memory's session span, with a small tolerance for event–mention mismatch; participant, entity, location, and topic cues provide auxiliary contextual coordinates. Full retrieval and compatibility details are provided in Appendix B.

3.4 EPISODIC BINDING PRESERVATION

Validity-constrained recall distinguishes memories using both semantic content and episodic coordinates. However, this benefit would be lost if the recalled memory were reduced again to content alone before downstream reasoning:

$$(x_i, e_i) \xrightarrow{\pi_{\mathcal{X}}} x_i. \tag{7}$$

Such a projection discards the same contextual distinctions that contextual reinstatement was designed to recover, allowing ambiguity to reappear after recall.

RAMEM therefore preserves the content–context binding of each recalled memory throughout downstream use. Let $\mathcal{M}_q^K = \text{TopK}_{\Pi_q}(\mathcal{M})$ denote the memories selected under the ordering in Eq. (6). The downstream reasoner operates on their bound representations:

$$y = G\Big(q,\, \{(x_i, e_i)\}_{m_i \in \mathcal{M}_q^K}\Big), \tag{8}$$

rather than on a content-only interface $G(q, \{x_i\})$. Thus, the reasoner retains not only *what* was recalled, but also the episodic conditions that distinguish the recalled trace from competing memories.

In our implementation, each recalled memory retains its available session span, event or mention time, participants, location, entities, and topic together with its content. These contextual coordinates remain attached to the corresponding memory during generation, while the recall ordering established by validity-constrained recall is preserved. In this way, RAMEM prevents the recovered episodic identity from being lost again at the recall–reasoning boundary.

Table 1: Performance on the LoCoMo benchmark. RaMem consistently improves answer quality across different backbone models and question categories.

Model	Method	MultiHop		Temporal		OpenDomain		SingleHop		Average	
		F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU
GPT-4o	LoCoMo	28.00	18.47	9.09	5.78	16.47	14.80	**61.56**	**54.19**	28.78	23.31
	ReadAgent	14.61	9.95	4.16	3.19	8.84	8.37	12.46	10.29	10.02	7.95
	MemoryBank	6.49	4.69	2.47	2.43	6.43	5.30	8.28	7.10	5.92	4.88
	MemGPT	30.36	22.83	17.29	13.18	12.24	11.87	40.16	36.35	25.01	21.06
	A-Mem	32.86	23.76	39.41	31.23	17.10	15.84	44.43	38.97	33.45	27.45
	LightMem	28.15	21.83	36.53	29.12	13.38	11.54	33.76	28.02	27.96	22.63
	Mem0	35.13	27.56	52.38	44.15	17.73	15.92	39.12	35.43	36.09	30.77
	SimpleMem	35.89	32.83	56.71	20.57	18.23	16.34	45.41	39.25	39.06	27.25
	RaMem (Ours)	**47.31**	**39.08**	**60.09**	**45.35**	**25.58**	**20.47**	52.87	47.08	**51.66**	**43.60**
GPT-4.1-mini	LoCoMo	25.02	21.62	12.04	10.63	19.05	17.07	18.68	15.87	18.70	16.30
	ReadAgent	6.48	5.60	5.31	4.23	7.66	6.62	9.18	7.91	7.16	6.09
	MemoryBank	5.00	4.68	5.94	4.78	5.16	4.52	5.72	4.86	5.46	4.71
	MemGPT	17.72	16.02	19.44	16.54	11.29	10.18	25.59	24.25	18.51	16.75
	A-Mem	25.06	17.32	51.01	44.75	13.22	14.75	41.02	36.99	32.58	28.45
	LightMem	24.96	21.66	20.55	18.39	19.21	17.68	33.79	29.66	24.63	21.85
	Mem0	30.14	27.62	48.91	44.82	16.43	14.94	41.30	36.17	34.20	30.89
	SimpleMem	43.46	**38.82**	58.62	**50.10**	19.76	18.04	51.12	43.53	43.24	37.62
	RaMem (Ours)	**43.73**	35.99	**61.57**	47.02	**26.35**	**21.73**	**58.14**	**51.48**	**54.23**	**45.86**
Qwen3-8b	LoCoMo	13.50	9.20	6.80	5.50	10.10	8.80	14.50	11.20	11.23	8.68
	ReadAgent	7.20	5.10	3.50	3.10	5.50	5.40	8.10	6.20	6.08	4.95
	MemoryBank	9.50	7.10	3.80	2.50	7.50	6.50	9.20	7.50	7.50	5.90
	MemGPT	14.20	9.80	5.50	4.20	12.50	10.80	11.50	9.10	10.93	8.48
	A-Mem	20.50	13.80	22.50	18.20	13.20	10.50	26.80	21.50	20.75	16.00
	LightMem	18.53	14.23	26.78	21.52	14.12	11.24	29.48	23.83	22.23	17.71
	Mem0	22.42	16.83	32.48	26.13	15.23	12.54	33.05	27.24	25.80	20.69
	SimpleMem	28.97	24.93	42.85	**36.49**	15.35	**13.90**	46.62	40.69	33.45	29.00
	RaMem (Ours)	**36.10**	**28.07**	**44.17**	32.72	**17.71**	13.79	**50.59**	**44.88**	**44.55**	**37.33**
Qwen2.5-3b	LoCoMo	4.61	4.29	3.11	2.71	4.55	5.97	7.03	5.69	4.83	4.67
	ReadAgent	2.47	1.78	3.01	3.01	5.57	5.22	3.25	2.51	3.58	3.13
	MemoryBank	3.60	3.39	1.72	1.97	6.63	6.58	4.11	3.32	4.02	3.82
	MemGPT	5.07	4.31	2.94	2.95	7.04	7.10	7.26	5.52	5.58	4.97
	A-Mem	12.57	9.01	**27.59**	**25.07**	7.12	7.28	17.23	13.12	16.13	13.62
	LightMem	16.43	11.39	6.92	4.56	8.06	7.23	18.28	15.24	12.42	9.61
	Mem0	16.89	11.54	8.52	6.23	10.24	8.82	16.47	12.43	13.03	9.76
	SimpleMem	17.03	11.87	21.47	19.50	12.52	10.19	20.90	18.01	17.98	14.89
	RaMem (Ours)	**21.74**	**15.17**	24.45	17.00	**13.68**	**11.41**	**26.95**	**22.72**	**24.65**	**19.44**

4 EXPERIMENTS

We evaluate whether RaMem improves long-term agent memory by mitigating context collapse through query-conditioned contextual reinstatement. Our experiments follow the SimpleMem (Liu et al., 2026) protocol on LoCoMo (Maharana et al., 2024) and LongMemEval (Wu et al., 2024). We evaluate across four backbones, GPT-4o, GPT-4.1-mini, Qwen2.5-3B, and Qwen3-8B, and report token-level F1 and BLEU-1 as the primary metrics. Full details and implementation are provided in Appendix C.1 for reproducibility and transparency.

4.1 MAIN RESULTS ON LONG-TERM MEMORY

Table 1 reports the main results on LoCoMo. RaMem achieves the best average F1 across all four backbones and consistently improves over SimpleMem, the strongest structured-memory baseline. Average F1 increases from 39.06 to 51.66 on GPT-4o, from 43.24 to 54.23 on GPT-4.1-mini, from 33.45 to 44.55 on Qwen3-8B, and from 17.98 to 24.65 on Qwen2.5-3B, with consistent BLEU improvements across backbones. Since SimpleMem already uses compressed memory units, multi-view indexing, and intent-aware retrieval, these gains suggest a complementary benefit of RaMem: contextual reinstatement preserves the episodic distinctions needed to recover the intended memory, rather than merely making memories compact and accessible. This benefit is not limited to temporal questions: substantial gains also appear in MultiHop, OpenDomain, and SingleHop. On GPT-4o, RaMem improves these categories by 11.42, 7.35, and 7.46 F1 points, respectively, while GPT-4.1-mini shows its strongest gains in OpenDomain and SingleHop. This pattern supports the intended role of contextual reinstatement: temporal and session information helps distinguish which episode

Table 2: Retrieval diagnostics on LoCoMo. RaMem improves both target-memory recall and ranking across backbone models. Lower values are better for CappedRank@10, HitOnlyRank, and GenFailGivenRetrieved.

Backbone	Method	R@1	R@3	R@5	R@10	MRR	CappedRank@10	HitOnlyRank	GenFailGivenRetrieved
GPT-4.1-mini	SimpleMem	0.3662	0.5636	0.6364	0.7221	0.4908	8.5292	8.3831	0.8336
	Ours	**0.4175**	**0.6305**	**0.6961**	**0.7890**	**0.5476**	**5.9844**	**5.6510**	**0.8109**
Qwen2.5-3B	SimpleMem	0.2740	0.4539	0.5234	0.6247	0.3968	10.0006	9.8928	0.9230
	Ours	**0.3054**	**0.4776**	**0.5809**	**0.6901**	**0.4313**	**8.0143**	**7.6847**	**0.9113**
Qwen3-8B	SimpleMem	0.2870	0.4903	0.5526	0.6617	0.4178	9.7390	9.6429	0.8693
	Ours	**0.3292**	**0.5435**	**0.6266**	**0.7351**	**0.4669**	**7.2532**	**6.9594**	**0.8340**

or event instance should be recalled when related memories compete. A more detailed analysis is provided in Appendix C.2.

4.2 Target Episode Retrieval Diagnostics

Final answer quality alone cannot reveal whether a memory system recovers the intended past episode, so we evaluate retrieval diagnostics using ground-truth memory entries. For each question, we record whether the answer-supporting memory appears in the retrieved set and where it is ranked. Table 2 shows that RaMem consistently improves target-memory retrieval across backbones. On GPT-4.1-mini, Recall@10 increases from 0.7221 to 0.7890, MRR improves from 0.4908 to 0.5476, and capped average rank drops from 8.5292 to 5.9844; Qwen2.5-3B and Qwen3-8B show similar Recall@10 gains of 6.54 and 7.34 points. The gains are stronger when contextual reinstatement is activated: on the temporal-triggered subset, GPT-4.1-mini improves Recall@10 from 0.6326 to 0.8140, while Qwen2.5-3B improves from 0.5256 to 0.6558. These results show that RaMem recovers the ground-truth memory more often and ranks it earlier, especially when the query refers to a specific episode and semantically related memories become strong competitors under content-centric recall. Generator failures can still occur after the correct memory is retrieved, but RaMem reduces this failure rate across all three backbones, suggesting that contextual reinstatement improves both access to the intended episode and the contextual structure available for downstream reasoning.

4.3 Context Collapse Analysis

To test whether RaMem mitigates context collapse, we analyze **context distractors**: retrieved memories related to the query or ground-truth memory but originating from an incompatible episodic context. We identify distractors using person, entity, topic, and keyword overlap for content relatedness, together with temporal or session mismatch for contextual incompatibility. We define a **context-confusable subset**, where SimpleMem retrieves at least one strict context distractor in its top-5 results, isolating cases in which episodically distinct memories become difficult to distinguish under content-centric recall. Figure 3 and Table 6 show that context collapse is frequent rather than incidental: SimpleMem ranks a context distractor first in 43.66%, 45.88%, and 54.14% of examples on GPT-4.1-mini, Qwen3-8B, and Qwen2.5-3B, respectively. These distractors share entities, topics, or keywords with the query but belong to episodic contexts incompatible with the target occurrence. RaMem reduces this interference, lowering D@1 to 0.3495, 0.3846, and 0.4796 across backbones. It also reduces RankGap by more than three points on GPT-4.1-mini and Qwen3-8B and by nearly three points on Qwen2.5-3B, while improving GT R@10 across backbones. These results show that RaMem suppresses wrong-episode interference without losing the answer-supporting memory, supporting our claim that contextual reinstatement restores the contextual distinctions needed to separate the target episode from semantically related alternatives.

4.4 Content–Context Binding Test

To test whether contextual reinstatement depends on the correct binding between memory content and its episodic coordinates, we conduct a context-shuffling intervention: memory content is kept unchanged, while episodic context fields are shuffled across entries within the same memory bank. This preserves both the amount and format of contextual information while selectively breaking the correspondence between a memory and the context in which it was encoded. We focus on temporal/session and full-context shuffling in the main text, with complete results in Appendix C.4.

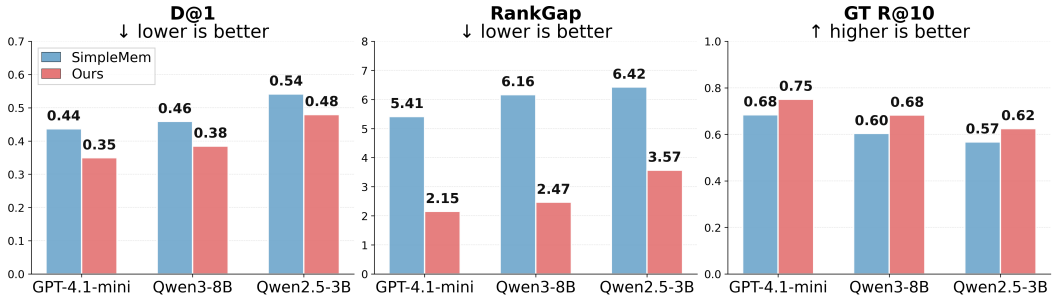


Figure 3: **Context collapse mitigation across backbones**. Each subplot corresponds to one backbone and compares SimpleMem with our method on three diagnostic metrics: D@1, RankGap, and GT R@10. Lower is better for D@1 and RankGap, while higher is better for GT R@10.

Table 3: Content–context binding test. Memory content is kept unchanged while selected episodic context fields are shuffled across entries, breaking their binding to the corresponding memories. We report both all verified questions and the temporal-triggered subset. Lower is better for D@1 and RankGap.

Backbone	Subset	Variant	F1 ↑	GT R@10 ↑	MRR ↑	D@1 ↓	RankGap ↓
GPT-4.1-mini	All Verified GT	Original	**0.5291**	**0.7550**	**0.5323**	**0.3428**	**2.0278**
		Shuffle temporal	0.5004	0.6744	0.4871	0.5753	5.2974
		Shuffle full context	0.5079	0.6638	0.4822	0.5872	5.3965
	Temporal-triggered	Original	**0.6009**	**0.8066**	**0.5249**	**0.2642**	**1.0404**
		Shuffle temporal	0.4985	0.2358	0.1796	0.8160	21.5029
		Shuffle full context	0.5180	0.2358	0.1762	0.8302	22.0116
Qwen3-8B	All Verified GT	Original	**0.4522**	**0.6925**	**0.4502**	**0.3702**	**2.1427**
		Shuffle temporal	0.4287	0.6204	0.4116	0.6518	5.7768
		Shuffle full context	0.4347	0.6157	0.4053	0.6631	5.7634
	Temporal-triggered	Original	**0.5317**	**0.7170**	**0.4271**	**0.2736**	**1.2350**
		Shuffle temporal	0.4081	0.1840	0.1250	0.8726	22.9595
		Shuffle full context	0.4134	0.1840	0.1242	0.8868	23.0952

As shown in Table 3, on all verified questions, temporal shuffling reduces GPT-4.1-mini F1 from 0.5291 to 0.5004 and GT R@10 from 0.7550 to 0.6744; on Qwen3-8B, F1 drops from 0.4522 to 0.4287 and GT R@10 from 0.6925 to 0.6204. The effect is much sharper on temporal-triggered questions: GT R@10 drops from 0.8066 to 0.2358 on GPT-4.1-mini and from 0.7170 to 0.1840 on Qwen3-8B, while D@1 rises from 0.2642 to 0.8160 and from 0.2736 to 0.8726, respectively. Full-context shuffling follows the same pattern, suggesting that the dominant degradation comes from breaking session-level temporal binding.

4.5 Hyper-parameter Analysis

We first study the sensitivity of RaMem to the temporal reinstatement window, which controls how much mismatch is allowed between event time and the time at which the event entered memory. As shown in Figure 4, a moderate buffer provides the best trade-off across GPT-4.1-mini and Qwen3-8B. Very narrow windows can miss contextually compatible memories whose mention time is slightly shifted, while overly broad windows admit more contextually incompatible memories and weaken contextual specificity. This supports our design intuition that temporal context should act as a calibrated episodic coordinate rather than a hard equality constraint or an unrestricted expansion window. Full results with additional diagnostics are provided in Appendix C.5. We also evaluate sensitivity to the retrieval budget in Figure 5, which controls how many memories are passed to the generator. RaMem consistently improves F1 and budget-aligned GT R@K across budgets, and notably matches or exceeds SimpleMem with half the retrieval budget ($K = 10$ vs. $K = 20$) on both GPT-4.1-mini and Qwen3-8B, indicating better recall ordering and context efficiency; full results are provided in Appendix C.6.

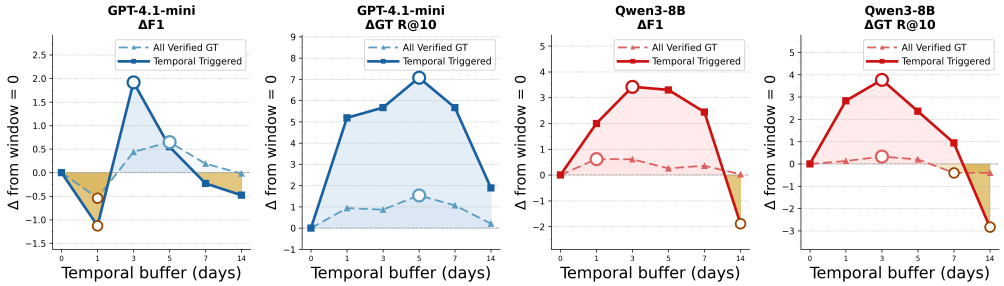


Figure 4: Sensitivity to the temporal reinstatement window. Each subplot shows how the temporal buffer affects F1 and GT R@10, reflecting the trade-off between tolerance and episodic specificity.

Table 4: Mechanism analysis on the temporal-triggered subset. We ablate the key mechanisms underlying contextual reinstatement and recall. Lower is better for D@1 and RankGap.

Backbone	Variant	F1 ↑	BLEU1 ↑	GT R@10 ↑	D@1 ↓	RankGap ↓
GPT-4.1-mini	Full	**0.5957**	**0.5319**	**0.8066**	**0.2736**	**1.0354**
	w/o Session Context	0.5596	0.4922	0.5708	0.5189	10.0492
	w/o Context-Aware Ranking	0.5693	0.5018	0.5708	0.5283	12.6332
	w/o Context-Preserved Generation	0.5626	0.4968	0.8066	0.2736	1.0354
Qwen3-8B	Full	**0.5226**	**0.4695**	**0.7217**	**0.2783**	**1.1685**
	w/o Session Context	0.4578	0.4097	0.4245	0.5849	11.9565
	w/o Context-Aware Ranking	0.4642	0.4162	0.4434	0.5755	14.7735
	w/o Context-Preserved Generation	0.4248	0.3567	0.7217	0.2783	1.1685

4.6 Mechanism Analysis

Since session-level contextual coordinates and validity-constrained ordering are most directly exercised on temporally grounded queries, we report the temporal-triggered subset in the main text and provide full ablations in Appendix C.8. Table 4 shows that session grounding and context-aware ranking are the main recall-side drivers. Without session context, GT R@10 drops from 0.8066 to 0.5708 on GPT-4.1-mini and from 0.7217 to 0.4245 on Qwen3-8B; D@1 roughly doubles and RankGap increases substantially. Removing context-aware ranking causes a similar degradation, indicating that RaMem must not only reinstate the contextual coordinates of the target episode, but also use them to constrain the ordering of competing memories. In contrast, removing context-preserved generation leaves retrieval metrics unchanged by design, but reduces F1 from 0.5957 to 0.5626 on GPT-4.1-mini and from 0.5226 to 0.4248 on Qwen3-8B. This shows that successful recall alone is insufficient: the recovered content–context binding must remain available during downstream reasoning.

4.7 Efficiency Analysis & Qualitative Case Studies

RaMem improves context efficiency without requiring a larger recall budget. As shown in Table 13, RaMem with $K = 10$ matches or exceeds SimpleMem with $K = 20$ on GPT-4.1-mini and Qwen3-8B, achieving better recall with roughly half the context tokens. Online-cost measurements are provided in Appendix C.9. **Case studies** in Appendix C.10 illustrate the same mechanism: SimpleMem retrieves semantically related memories from incorrect episodes, whereas RaMem promotes memories whose episodic coordinates are compatible with the context reinstated from the query.

5 Conclusion

In this paper, we presented RaMem, a contextual reinstatement framework for long-term agent memory. RaMem addresses context collapse by recovering query-relevant episodic coordinates, conditioning recall on contextual compatibility, and preserving content–context binding during downstream reasoning. Experiments across multiple backbones on LoCoMo show consistent gains in answer quality, ground-truth memory retrieval, and robustness to wrong-episode interference. These results highlight contextual reinstatement as an important principle for reliable long-term recall.

AI USE STATEMENT

Generative AI tools were used to assist with language refinement, proofreading, and presentation of the manuscript. They were not used to determine the research questions, design the proposed method, conduct experiments, or produce the reported empirical results. All AI-assisted material was carefully reviewed and edited by the authors for technical correctness, consistency with the experiments, and appropriate scholarly presentation. The authors take full responsibility for all content, claims, and artifacts in the final manuscript.

ETHICS STATEMENT

This work studies long-term memory mechanisms for LLM-based agents using publicly released conversational memory benchmarks. It does not involve human-subject experimentation or the collection of private user data. The proposed method focuses on memory representation, contextual recall, and evaluation in benchmark settings. We follow the usage conditions of the employed datasets and models and do not introduce additional personally identifiable or sensitive user information.

REPRODUCIBILITY STATEMENT

We provide the formulation and core design of RAMEM in the main paper, with additional implementation details, memory construction procedures, retrieval configurations, prompts, evaluation protocols, and ablation settings in the appendix. Anonymous source code and supporting materials for reproducing the reported experiments are available at `https://anonymous.4open.science/r/ramem_repo-F4C4/`.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2023.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Zouying Cao, Jiaji Deng, Li Yu, Weikang Zhou, Zhaoyang Liu, Bolin Ding, and Hai Zhao. Remember me, refine me: A dynamic procedural memory framework for experience-driven agent evolution. *arXiv preprint arXiv:2512.10696*, 2025.

Ching Chang, Yidan Shi, Defu Cao, Wei Yang, Jeehyun Hwang, Haixin Wang, Jiacheng Pang, Wei Wang, Yan Liu, Wen-Chih Peng, et al. A survey of reasoning and agentic systems in time series with large language models. *arXiv preprint arXiv:2509.11575*, 2025.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

Yiqun Chen, Qi Liu, Yi Zhang, Weiwei Sun, Xinyu Ma, Wei Yang, Daiting Shi, Jiaxin Mao, and Dawei Yin. Tourrank: Utilizing large language models for documents ranking with a tournament-inspired strategy. In *Proceedings of the ACM on Web Conference 2025*, pp. 1638–1652, 2025.

Yiqun Chen, Jinyuan Feng, Wei Yang, Meizhi Zhong, Zhengliang Shi, Rui Li, Xiaochi Wei, Yan Gao, Yi Wu, Yao Hu, et al. Self-compression of chain-of-thought via multi-agent reinforcement learning. *arXiv preprint arXiv:2601.21919*, 2026a.

Yiqun Chen, Wei Yang, Erhan Zhang, Shijie Wang, Qi Liu, Zechun Niu, Bin Zhang, Haitao Li, Rui Li, Lingyong Yan, et al. Unitymas-o: A general rl optimization framework for llm-based multi-agent systems. *arXiv preprint arXiv:2605.26646*, 2026b.

Mingyue Cheng, Jie Ouyang, Shuo Yu, Ruiran Yan, Yucong Luo, Zirui Liu, Daoyu Wang, Qi Liu, and Enhong Chen. Agent-r1: Training powerful llm agents with end-to-end reinforcement learning, 2025. URL `https://arxiv.org/abs/2511.14460`.

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

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019. URL https://arxiv.org/abs/1904.10509.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL https://aclanthology.org/P19-1285/.

Pengfei Du. Memory for autonomous llm agents:mechanisms, evaluation, and emerging frontiers, 2026. URL https://arxiv.org/abs/2603.07670.

Xingbo Du, Loka Li, Duzhen Zhang, and Le Song. Memr ^{3}: Memory retrieval via reflective reasoning for llm agents. *arXiv preprint arXiv:2512.20237*, 2025a.

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z. Pan. Rethinking memory in llm based agents: Representations, operations, and emerging topics, 2025b. URL https://arxiv.org/abs/2505.00675.

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

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

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Hao Gu, Rui Zhong, Yu Xia, Wei Yang, Chi Lu, Peng Jiang, and Kun Gai. R 4ec: A reasoning, reflection, and refinement framework for recommendation systems. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pp. 411–421, 2025.

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

Marc W Howard and Michael J Kahana. A distributed representation of temporal context. *Journal of mathematical psychology*, 46(3):269–299, 2002.

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

Alexis Huet, Zied Ben Houidi, and Dario Rossi. Episodic memories generation and evaluation benchmark for large language models, 2025. URL https://arxiv.org/abs/2501.13121.

Zixi Jia, Qinghua Liu, Hexiao Li, Yuyan Chen, and Jiqiang Liu. Evaluating the long-term memory of large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19759–19777, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1014. URL https://aclanthology.org/2025.findings-acl.1014/.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 13358–13376, 2023a.

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

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.09516.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25972–25981, 2025.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering, 2020. URL https://arxiv.org/abs/2004.04906.

Christopher A Kurby and Jeffrey M Zacks. Segmentation in the perception and memory of events. *Trends in cognitive sciences*, 12(2):72–79, 2008.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts. *arXiv preprint arXiv:2402.09727*, 2024.

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

Li Li, Chenwei Wang, You Qin, Wei Ji, and Renjie Liang. Biased-predicate annotation identification via unbiased visual predicate representation. In *ACM MM*, pp. 4410–4420. Association for Computing Machinery, 2023. ISBN 9798400701085. doi: 10.1145/3581783.3611847. URL https://doi.org/10.1145/3581783.3611847.

Li Li, Wei Ji, Yiming Wu, Mengze Li, You Qin, Lina Wei, and Roger Zimmermann. Panoptic scene graph generation with semantics-prototype learning. *AAAI*, 38(4):3145–3153, Mar. 2024a. doi: 10.1609/aaai.v38i4.28098.

Shawn Li and Yue Zhao. The autonomy tax: Defense training breaks llm agents, 2026. URL https://arxiv.org/abs/2603.19423.

Shawn Li, Peilin Cai, Yuxiao Zhou, Zhiyu Ni, Renjie Liang, You Qin, Yi Nian, Zhengzhong Tu, Xiyang Hu, and Yue Zhao. Secure on-device video ood detection without backpropagation. In *ICCV*, October 2025a.

Shawn Li, Huixian Gong, Hao Dong, Tiankai Yang, Zhengzhong Tu, and Yue Zhao. Dpu: Dynamic prototype updating for multimodal out-of-distribution detection. In *CVPR*, pp. 10193–10202, June 2025b.

Shawn Li, Chenxiao Yu, Zhiyu Ni, Hao Li, Charith Peris, Chaowei Xiao, and Yue Zhao. Defenses against prompt attacks learn surface heuristics. In *ACL*, 2026a.

Shawn Li, Chenxiao Yu, Han Wang, Wei Yang, Ryan Rossi, Franck Dernoncourt, Xiyang Hu, Philip Yu, Chaowei Xiao, Huan Zhang, et al. Fortis: Benchmarking over-privilege in agent skills. *arXiv preprint arXiv:2605.09163*, 2026b.

Shixuan Li, Wei Yang, Peiyu Zhang, Xiongye Xiao, Defu Cao, Yuehan Qin, Xiaole Zhang, Yue Zhao, and Paul Bogdan. Climatellm: Efficient weather forecasting via frequency-aware large language models. *arXiv preprint arXiv:2502.11059*, 2025c.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. Retrieval augmented generation or long-context LLMs? a comprehensive study and hybrid approach. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 881–893, Miami, Florida, US, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.66. URL https://aclanthology.org/2024.emnlp-industry.66/.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Simplemem: Efficient lifelong memory for llm agents. *arXiv preprint arXiv:2601.02553*, 2026.

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

WenTao Liu, Ruohua Zhang, Aimin Zhou, Feng Gao, and JiaLi Liu. Echo: A large language model with temporal episodic memory, 2025. URL https://arxiv.org/abs/2502.16090.

Wenquan Ma, Jiayan Nan, Wenlong Wu, and Yize Chen. What deserves memory: Adaptive memory distillation for llm agents. *arXiv e-prints*, pp. arXiv–2508, 2025.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 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)*, pp. 13851–13870, 2024.

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

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

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023. URL https://arxiv.org/abs/2304.03442.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models, 2026. URL https://arxiv.org/abs/2309.00071.

Heng Ping, Arijit Bhattacharjee, Peiyu Zhang, Shixuan Li, Wei Yang, Anzhe Cheng, Xiaole Zhang, Jesse Thomason, Ali Jannesari, Nesreen Ahmed, et al. Verimoa: A mixture-of-agents framework for spec-to-hdl generation. *arXiv preprint arXiv:2510.27617*, 2025a.

Heng Ping, Shixuan Li, Peiyu Zhang, Anzhe Cheng, Shukai Duan, Nikos Kanakaris, Xiongye Xiao, Wei Yang, Shahin Nazarian, Andrei Irimia, et al. Hdlcore: A training-free framework for mitigating hallucinations in llm-generated hdl. In *2025 IEEE International Conference on LLM-Aided Design (ICLAD)*, pp. 108–116. IEEE, 2025b.

Heng Ping, Peiyu Zhang, Shixuan Li, Wei Yang, Anzhe Cheng, Shukai Duan, Xiaole Zhang, and Paul Bogdan. Coevo: Co-evolutionary framework for joint functional correctness and ppa optimization in llm-based rtl generation. *arXiv preprint arXiv:2604.15001*, 2026a.

Heng Ping, Peiyu Zhang, Zhenkun Wang, Shixuan Li, Anzhe Cheng, Wei Yang, Paul Bogdan, and Shahin Nazarian. Poet: Power-oriented evolutionary tuning for llm-based rtl ppa optimization. *arXiv preprint arXiv:2603.19333*, 2026b.

Mathis Pink, Qinyuan Wu, Vy Ai Vo, Javier Turek, Jianing Mu, Alexander Huth, and Mariya Toneva. Position: Episodic memory is the missing piece for long-term llm agents, 2025. URL https://arxiv.org/abs/2502.06975.

Yuehan Qin, Li Li, Linxin Song, Wei Yang, Jiate Li, Yuqing Yang, and Yue Zhao. Memory retrieval for changing preferences. *arXiv preprint arXiv:2606.02976*, 2026.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: a temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*, 2025.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009. ISSN 1554-0669. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Li Shawn, Jiashu Qu, Linxin Song, Yuxiao Zhou, Yuehan Qin, Tiankai Yang, and Yue Zhao. Treble counterfactual VLMs: A causal approach to hallucination. In *EMNLP*, pp. 18423–18434, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Yiheng Shu, Saisri Padmaja Jonnalagedda, Xiang Gao, Bernal Jiménez Gutiérrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. Remem: Reasoning with episodic memory in language agent, 2026. URL https://arxiv.org/abs/2602.13530.

Aditi Singh, Abul Ehtesham, Saket Kumar, Tala Talaei Khoei, and Athanasios V. Vasilakos. Agentic retrieval-augmented generation: A survey on agentic rag, 2026. URL https://arxiv.org/abs/2501.09136.

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

Zhenheng Tang, Xin He, Tiancheng Zhao, Fanjunduo Wei, Xiang Liu, Peijie Dong, Qian Wang, Qi Li, Huacan Wang, Ronghao Chen, Sen Hu, Weidong Guo, Yu Xu, Haolan Chen, Kunfeng Lai, Kaiyong Zhao, Keyan Ding, Ivor W. Tsang, Yew-Soon Ong, Bo Li, and Xiaowen Chu. Llm agent memory: A survey from a unified representation–management perspective. *Preprints*, March 2026. doi: 10.20944/preprints202603.0359.v2. URL https://doi.org/10.20944/preprints202603.0359.v2.

Endel Tulving. Episodic memory: From mind to brain. *Annual review of psychology*, 53(1):1–25, 2002.

Endel Tulving and Donald M Thomson. Encoding specificity and retrieval processes in episodic memory. *Psychological review*, 80(5):352, 1973.

Bing Wang, Xinnian Liang, Jian Yang, Hui Huang, Shuangzhi Wu, Peihao Wu, Lu Lu, Zejun Ma, and Zhoujun Li. Enhancing large language model with self-controlled memory framework. *arXiv preprint arXiv:2304.13343*, 2023.

Piaohong Wang, Motong Tian, Jiaxian Li, Yuan Liang, Yuqing Wang, Qianben Chen, Tiannan Wang, Zhicong Lu, Jiawei Ma, Yuchen Eleanor Jiang, et al. O-mem: Omni memory system for personalized, long horizon, self-evolving agents. *arXiv preprint arXiv:2511.13593*, 2025a.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning, 2025b. URL https://arxiv.org/abs/2504.20073.

Muyan Weng, Defu Cao, Wei Yang, Yashaswi Sharma, and Yan Liu. Temporalbench: A benchmark for evaluating llm-based agents on contextual and event-informed time series tasks. *arXiv preprint arXiv:2602.13272*, 2026.

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

Yu Xia, Rui Zhong, Zeyu Song, Wei Yang, Junchen Wan, Qingpeng Cai, Chi Lu, and Peng Jiang. Trackrec: Iterative alternating feedback with chain-of-thought via preference alignment for recommendation. In *International Conference on Database Systems for Advanced Applications*, pp. 388–404. Springer, 2026.

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

Chang Yang, Chuang Zhou, Yilin Xiao, Su Dong, Luyao Zhuang, Yujing Zhang, Zhu Wang, Zijin Hong, Zheng Yuan, Zhishang Xiang, Shengyuan Chen, Huachi Zhou, Qinggang Zhang, Ninghao Liu, Jinsong Su, Xinrun Wang, Yi Chang, and Xiao Huang. Graph-based agent memory: Taxonomy, techniques, and applications, 2026a. URL https://arxiv.org/abs/2602.05665.

Wei Yang and Jesse Thomason. Learning to deliberate: Meta-policy collaboration for agentic llms with multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29820–29828, 2026.

Wei Yang, Jiacheng Pang, Shixuan Li, Paul Bogdan, Stephen Tu, and Jesse Thomason. Maestro: Learning to collaborate via conditional listwise policy optimization for multi-agent llms. *arXiv preprint arXiv:2511.06134*, 2025a.

Wei Yang, Muyan Weng, Jiacheng Pang, Defu Cao, Heng Ping, Peiyu Zhang, Shixuan Li, Yue Zhao, Qiang Yang, Mengdi Wang, et al. Toward evolutionary intelligence: Llm-based agentic systems with multi-agent reinforcement learning. *Available at SSRN 5819182*, 2025b.

Wei Yang, Defu Cao, Jiacheng Pang, Muyan Weng, and Yan Liu. Adaptive collaboration with humans: Metacognitive policy optimization for multi-agent llms with continual learning. *arXiv preprint arXiv:2603.07972*, 2026b.

Wei Yang, Shixuan Li, Heng Ping, Peiyu Zhang, Paul Bogdan, and Jesse Thomason. Auditing multi-agent llm reasoning trees outperforms majority vote and llm-as-judge. *arXiv preprint arXiv:2602.09341*, 2026c.

Zixuan Yang, Yiqun Chen, Wei Yang, Erhan Zhang, Zihan Shen, Xiaochi Wei, Yan Gao, Yi Wu, Yao Hu, and Jiaxin Mao. Tournament-grpo: Group-wise tournament rewards for reinforcement learning in open-ended long-form generation. *arXiv preprint arXiv:2605.26958*, 2026d.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Wen Ye, Jinbo Liu, Defu Cao, Wei Yang, and Yan Liu. When llm meets time series: Can llms perform multi-step time series reasoning and inference. *arXiv preprint arXiv:2509.01822*, 2025.

Wen Ye, Wei Yang, Defu Cao, Yizhou Zhang, Lumingyuan Tang, Jie Cai, and Yan Liu. Ts-reasoner: Domain-oriented time series inference agents for reasoning and automated analysis. *Transactions on Machine Learning Research*, 2026.

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

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. *arXiv preprint arXiv:2601.01885*, 2026.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

Erhan Zhang, Yiqun Chen, Zechun Niu, Wei Yang, Xiaochi Wei, Yan Gao, Yi Wu, Yao Hu, and Jiaxin Mao. Oases: Outcome-aligned search-evaluation co-training for agentic search. *arXiv preprint arXiv:2604.03675*, 2026a.

Guilin Zhang, Wei Jiang, Xiejiashan Wang, Aisha Behr, Kai Zhao, Jeffrey Friedman, Xu Chu, and Amine Anoun. Adaptive memory admission control for llm agents. *arXiv preprint arXiv:2603.04549*, 2026b.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Trans. Inf. Syst.*, 43(6), September 2025. ISSN 1046-8188. doi: 10.1145/3748302. URL https://doi.org/10.1145/3748302.

Rui Zhao, Rui Zhong, Haoran Zheng, Wei Yang, Chi Lu, Beihong Jin, Peng Jiang, and Kun Gai. Hierarchical sequence id representation of large language models for large-scale recommendation systems. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 641–650, 2025.

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

APPENDIX

A RELATED WORK

A.1 LLM AGENTS AND LONG-TERM MEMORY

The rapid development of large language models has been driven by advances in large-scale pretraining, which have enabled strong general-purpose capabilities across understanding (Chang et al., 2025; Li et al., 2025c), reasoning (Chen et al., 2026a; 2025; Xia et al., 2026), coding (Ping et al., 2026a;b), recommendation (Gu et al., 2025; Zhao et al., 2025), and other domains (Ping et al., 2025b; Weng et al., 2026; Ye et al., 2026). Building on these capabilities, recent work has increasingly studied LLM-based agents that can interact with external tools, environments, users, and other agents to solve complex tasks beyond single-turn text generation (Yang et al., 2026d). In particular, multi-agent LLM systems (Chen et al., 2026b; Yang & Thomason, 2026; Yang et al., 2025b) organize multiple specialized agents into collaborative workflows (Yang et al., 2026b; 2025a), and have been widely explored in settings such as question answering (Zhang et al., 2026a), decision making (Yang et al., 2026c), and other applications (Li et al., 2026b; Ping et al., 2025a; Ye et al., 2025). As these agentic systems move from isolated tasks to long-horizon interactions, persistent memory becomes a key mechanism for maintaining continuity, reusing prior experience, and grounding future decisions in past context.

Long-term memory has become an important mechanism for extending LLM agents beyond the limits of a single context window (Tang et al., 2026; Zhang et al., 2025; Hu et al., 2025; Shawn et al., 2025; Li et al., 2026a; Du, 2026; Qin et al., 2026). A natural early direction is to maintain external memory buffers or virtual context systems, where past interactions can be paged, summarized, or retrieved when the model needs them (Park et al., 2023; Shinn et al., 2023). MemGPT (Packer et al., 2024) frames this problem through an operating-system analogy, allowing agents to move information between active context and external storage. MemoryBank (Zhong et al., 2024) stores and updates long-term user-related memories for personalized interaction. SCM (Wang et al., 2023) and MemoryOS (Kang et al., 2025) further explore controller-style memory streams and modular memory management. These systems establish the basic architecture of non-parametric agent memory, but they often keep memory close to raw dialogue records or summaries, which can preserve redundancy and leave referential or temporal ambiguity unresolved.

A second line of work introduces more structured memory representations (Du et al., 2025b; Yang et al., 2026a). Zep (Rasmussen et al., 2025) represents evolving conversational information with a temporal knowledge graph, while A-Mem (Xu et al., 2025) organizes memory in a self-evolving, note-linking structure inspired by Zettelkasten. Mem0 (Chhikara et al., 2025) extracts compact persistent facts from conversations and further extends them with graph-based memory to model relational structure. LightMem (Fang et al., 2025) pursues a lightweight architecture inspired by classic memory models, and O-Mem (Wang et al., 2025a) studies broader omni-memory organization for long-horizon agents. These methods move beyond passive context retention by adding structure to memory storage and update. Their main focus, however, remains on how to represent, consolidate, and maintain useful information over time.

More recent work has shifted from memory storage toward memory distillation and adaptive management (Xu et al., 2025; Tan et al., 2025; Li & Zhao, 2026; Zhang et al., 2026b). SimpleMem (Liu et al., 2026) compresses dialogue into self-contained memory units, indexes them through semantic, lexical, and symbolic views, and uses intent-aware retrieval to reduce token cost. NEMORI (Nan et al., 2025) asks what deserves to be stored by relating retention to predictability and prediction error. AgeMem (Yu et al., 2026) exposes long-term and short-term memory operations as actions and optimizes memory behavior through reinforcement learning. MemAgent (Yu et al., 2025) also treats memory update as a trainable process for long-context reading. Together, these works show a clear progression from storing history, to structuring memory, to learning or adapting memory operations. Our work follows this progression but studies a different point in the memory pipeline: after memories have been stored, retrieval can still fail when semantically similar memories from different episodes become difficult to distinguish (Maharana et al., 2024).

A.2 Context Management and Agentic Retrieval

The problem of using long histories is closely related to long-context modeling and retrieval-augmented generation (Beltagy et al., 2020; Li et al., 2024a; 2025b; Zaheer et al., 2020; Li et al., 2024b). Long-context LLMs and position-extrapolation methods attempt to expand the amount of text that can be directly processed by the model (Chen et al., 2023; Peng et al., 2026), while sparse attention, recurrent architectures, and state-space models seek more efficient sequence processing (Child et al., 2019; Li et al., 2025a; 2023; Dai et al., 2019; Gu & Dao, 2024). Even with longer contexts, empirical work on the "lost-in-the-middle" effect shows that simply placing more information in the prompt does not guarantee reliable use of relevant evidence (Liu et al., 2024). Prompt compression methods such as LLMLingua (Jiang et al., 2023a) reduce context cost by shortening inputs, while retrieval-augmented generation decouples knowledge storage from generation by retrieving a smaller evidence set at inference time (Lewis et al., 2020). Follow-up RAG systems, including Self-RAG (Asai et al., 2023), active retrieval methods (Jiang et al., 2023b), GraphRAG (Edge et al., 2024), and LightRAG (Guo et al., 2024), improve retrieval control, graph-structured evidence aggregation, or query-dependent evidence use.

Although these techniques improve context access, long-term conversational memory introduces a distinct challenge (Huet et al., 2025; Pink et al., 2025). Unlike static documents or knowledge bases, dialogue histories contain recurring people, places, topics, and events across multiple sessions (Maharana et al., 2024; Jia et al., 2025). A retrieved memory may therefore be highly relevant in wording but still refer to the wrong episode (Liu et al., 2025; Shu et al., 2026). This issue is only partially addressed by standard dense retrieval, lexical retrieval, or symbolic filtering(Karpukhin et al., 2020; Robertson & Zaragoza, 2009). ReadAgent (Lee et al., 2024) compresses very long contexts into gist memories, which improves scalability but does not directly resolve episode-level ambiguity. SimpleMem (Liu et al., 2026) uses multi-view retrieval to combine semantic, lexical, and symbolic access paths, but its retrieval objective is still primarily organized around retrieving compact relevant memories under a token budget.

A parallel trend makes retrieval itself more agentic (Singh et al., 2026). ReAct (Yao et al., 2022) combines reasoning and acting, Reflexion (Shinn et al., 2023) stores verbal feedback from past trials, and Self-RAG (Asai et al., 2023) lets the model decide when retrieval or critique is needed. More recent agent-training frameworks such as Search-R1 (Jin et al., 2025), Agent-R1 (Cheng et al., 2025), and RAGEN (Wang et al., 2025b) extend this idea to multi-turn retrieval or tool-use settings. For long-term conversational memory, MemR^3 (Du et al., 2025a) introduces a controller that routes among retrieve, reflect, and answer actions while maintaining an explicit evidence-gap state. These methods address when to retrieve more evidence and how to control multi-step evidence collection. Our work is complementary: instead of emphasizing retrieval depth or closed-loop control, we focus on the validity of retrieved memories under the query's episodic context.

B Method

This appendix provides additional details for RaMem, complementing the formulation in the main paper with the concrete instantiation of each mechanism. We describe the context-bound memory representation, grounding procedure for conservative contextual reinstatement, implementation of validity-constrained recall, and preservation of episodic binding during downstream reasoning, together with the supporting retrieval and generation details required for reproducibility.

B.1 Additional Details on Context-Bound Memory Representation

As defined in Section 3.1, each memory is represented as a context-bound trace $m_i = (x_i, e_i) \in \mathcal{X} \times \mathcal{E}$. In our implementation, the episodic coordinates are

$$e_i = \left\{ \tau_i^{\text{event}}, \tau_i^{\text{mention}}, \tau_i^{\text{session}}, p_i, \ell_i, u_i, z_i \right\}, \tag{9}$$

where τ_i^{event} denotes the time of the described event when inferable, τ_i^{mention} an explicit temporal reference, $\tau_i^{\text{session}} = [s_i^{\text{start}}, s_i^{\text{end}}]$ the interaction span in which the memory was observed, and p_i, ℓ_i, u_i, and z_i denote participants, location, entities, and topic, respectively.

We distinguish event time from interaction time because an event may be discussed after it occurs. Accordingly, τ_i^{event} describes when the remembered event happened, whereas τ_i^{session} records when

that information entered the interaction history. The latter provides a stable episodic coordinate for contextual reinstatement, while the remaining fields retain complementary contextual distinctions when available. These coordinates remain bound to the corresponding memory content rather than being treated as detached metadata.

B.2 Grounding Details for Conservative Contextual Reinstatement

As defined in Section 3.2, RAMEM reconstructs a partial contextual state $\hat{c}_q = \{\hat{\tau}_q, \hat{p}_q, \hat{\ell}_q, \hat{u}_q, \hat{z}_q\}$ from the current query. These coordinates correspond to temporal, participant, location, entity, and episode or topic cues, respectively. The reconstruction is intentionally partial: only coordinates supported by the query are instantiated, while unspecified dimensions remain unconstrained.

In our implementation, session-level temporal context provides the primary explicit reinstatement coordinate. Absolute or relative temporal expressions are resolved against the interaction timeline to obtain a grounded interval $\hat{\tau}_q$. Temporal reinstatement is activated only when the original query contains a detectable temporal cue and that cue can be resolved to a valid range; otherwise, no temporal constraint is imposed. This prevents unsupported temporal inferences from excluding the target episode.

Participant, entity, location, and topic information provide auxiliary contextual cues when available. They are retained as part of the reinstated context but are not treated as unconditional hard constraints. Thus, contextual reinstatement remains selective: the query contributes only the episodic coordinates that can be grounded reliably, while unresolved dimensions fall back to content-based recall.

B.3 Instantiation of Validity-Constrained Recall

Section 3.3 defines recall through contextual validity $V_q(m_i) = C(e_i, \hat{c}_q)$ and a conservative validity-first ordering. Here we describe its concrete instantiation. For content relevance, RAMEM combines complementary semantic and lexical access paths. Let $\mathcal{C}_{\text{sem}}(q)$ and $\mathcal{C}_{\text{lex}}(q)$ denote their ranked candidate lists; we obtain the broad content-oriented set as $\mathcal{C}_{\text{content}}(q) = \text{RRF}(\mathcal{C}_{\text{sem}}(q), \mathcal{C}_{\text{lex}}(q))$. When contextual reinstatement is grounded, the same memory bank is restricted by the reinstated coordinates to obtain context-compatible candidates $\mathcal{C}_{\text{ctx}}(q)$. These candidates are prioritized before the content-oriented list, with stable deduplication; when contextual grounding is unavailable, recall reduces to $\mathcal{C}_{\text{content}}(q)$. Thus, contextual compatibility changes recall priority without eliminating relevance-based fallback.

In the current implementation, session-level time provides the primary explicit instantiation of $C(e_i, \hat{c}_q)$. Given a grounded query interval $\hat{\tau}_q = [t_q^{\text{start}}, t_q^{\text{end}}]$ and memory session span $\tau_i^{\text{session}} = [s_i^{\text{start}}, s_i^{\text{end}}]$, temporal compatibility is

$$V_q^{\tau}(m_i) = \mathbb{I}\left[s_i^{\text{start}} \leq t_q^{\text{end}} + \delta \ \wedge \ s_i^{\text{end}} \geq t_q^{\text{start}} - \delta\right], \tag{10}$$

where δ provides tolerance for mismatch between event time and the session in which the event was mentioned. Within the compatible set, memories are ordered by content relevance together with temporal proximity to the reinstated interval.

Participant, entity, location, and topic coordinates provide additional contextual cues when available, while temporal/session grounding remains the primary explicit validity signal in our experiments. This implementation realizes the validity-first recall principle without requiring contextual cues to act as irreversible filters.

B.4 Implementation of Episodic Binding Preservation

As described in Section 3.4, RAMEM preserves the content–context binding of each recalled memory during downstream reasoning. In our implementation, every selected memory retains its content together with the available session span, event or mention time, participants, location, entities, and topic. These contextual coordinates remain attached to the corresponding memory rather than being provided as detached global metadata.

The recalled memories are presented to the downstream model in the order established by validity-constrained recall, with each item's episodic coordinates kept locally associated with its content. This

preserves the contextual distinctions recovered during reinstatement without introducing an additional memory transformation, ensuring that the target episode remains identifiable at the recall–reasoning boundary.

C EXPERIMENTS

C.1 EXPERIMENTAL SETUP

Dataset. We follow the SimpleMem evaluation protocol and evaluate on LoCoMo (Maharana et al., 2024), a long-term conversational memory benchmark with multi-session dialogue histories and questions that require recovering information from prior interactions. The benchmark contains recurring entities, interleaved topics, and temporal shifts across sessions, making it suitable for evaluating whether a memory system can identify the correct episode rather than only retrieve related fragments. Following our evaluation split, we use samples from the dataset, and evaluate questions from four categories: multi-hop, temporal, open-domain, and single-hop questions. For analyses that require a verified answer-supporting memory, we use the subset with a reliable single ground-truth memory entry. For LongMemEval (Wu et al., 2024), we follow the standard accuracy-style evaluation used in SimpleMem, where a judge model compares the generated answer with the reference answer and labels the response as correct or incorrect based on semantic and temporal consistency.

Baselines and Backbones. For broad comparison, we use the same representative memory baselines reported in SimpleMem, including the full-history LoCoMo setting (Maharana et al., 2024), ReadAgent (Lee et al., 2024), MemoryBank (Zhong et al., 2024), MemGPT (Packer et al., 2024), A-Mem (Xu et al., 2025), LightMem (Fang et al., 2025), Mem0 (Chhikara et al., 2025), and SimpleMem (Liu et al., 2026). We reproduced SimpleMem with its official repository and obtained comparable results, so we report baseline numbers from the SimpleMem paper to maintain consistency with its evaluation protocol. These methods cover full-context prompting, gist-style reading memory, persistent user memory, virtual memory, structured or graph-based memory, lightweight retrieval, and semantic-compression-based memory. Following SimpleMem, we evaluate across both closed-source and open-source backbones: GPT-4o, GPT-4.1-mini, Qwen2.5-3B, and Qwen3-8B. For diagnostic analyses that require retrieved memory traces, we compare RaMem against a SimpleMem-style baseline reproduced inside our current implementation, so that both methods share the same serving stack, memory database, and evaluation scripts.

Metrics. For answer quality, we report token-level F1 and BLEU-1, following the LoCoMo evaluation protocol. We also report category-level results for multi-hop, temporal, open-domain, and single-hop questions. To verify whether performance gains come from better memory access, we compute retrieval diagnostics using verified ground-truth memory entries, including Recall@K, MRR, capped ground-truth rank, hit-only rank, and generator failure conditioned on the ground-truth memory being retrieved. For context-collapse analysis, we measure the rate of context distractors at top ranks, average distractor load in the top-5, and the rank gap between the ground-truth memory and the strongest context distractor. Unless otherwise stated, retrieval diagnostics use $K_{\max} = 10$.

Implementation Details. We match the SimpleMem (Liu et al., 2026) implementation as closely as possible. Dialogue histories are segmented into overlapping windows, and memory entries are stored in LanceDB with multi-view indexing. Dense retrieval uses Qwen3-Embedding-0.6B with 1024-dimensional embeddings, lexical retrieval uses BM25-style sparse indexing, and structured retrieval uses metadata fields such as timestamps, persons, entities, locations, and session-level context, while topic information is preserved for generation-time grounding. Retrieval budgets and windowing parameters are fixed within each compared setting so that methods are evaluated under the same memory construction and evidence-selection protocol. RaMem keeps the same memory construction and indexing backbone, but replaces the original retrieval stage with contextual reinstatement: it grounds recall conditions from the query, activates temporal retrieval only when reliable cues are available, applies a default ± 5 day temporal buffer, and preserves structured fields such as session, time, location, persons, entities, and topic in the generation prompt. OpenAI models are accessed through OpenAI-compatible APIs, while Qwen models are served locally with vLLM when applicable. All reported comparisons use the same generator backbone within each setting. For LongMemEval-S, we use a benchmark-specific adapter on top of the core RaMem retrieval pipeline.

The adapter preserves session-level memory construction and, when needed, applies exact-match fallback over generated memory entries and raw-haystack fallback over salient prior messages; these adapter-specific fallbacks are used only for LongMemEval-S evaluation and are separate from the core RaMem method.

Main Prompts. We provide the main prompts used in RaMem for memory construction, query-side recall condition induction, retrieval planning, and final answer generation. Runtime variables are shown in braces, e.g., {query} and {context_str}.

Memory Extraction Prompt

System. You are a professional information extraction assistant, skilled at extracting structured, unambiguous information from conversations. You must output valid JSON format.

User. Extract all valuable information from the following dialogues and convert them into structured memory entries.

Input. {dialogue_text}

Requirements.

1. Generate enough memory entries to capture all useful information in the dialogues.
2. Do not use pronouns or unresolved relative time expressions.
3. Each lossless_restatement must be complete, independent, and understandable.
4. Extract structured fields: keywords, timestamp, location, persons, entities, and topic.
5. Create separate entries for separate facts, preferences, plans, events, locations, relationships, work details, and media references.
6. Return valid JSON only.

Output Format.

```
{
"memory_entries": [
{
"lossless_restatement": "Complete unambiguous
restatement.",
"keywords": ["keyword1", "keyword2"],
"timestamp": "YYYY-MM-DDTHH:MM:SS or null",
"location": "location name or null",
"persons": ["name1", "name2"],
"entities": ["entity1", "entity2"],
"topic": "topic phrase"
}
]
}
```

Constraint. Return only the JSON object.

Query Analysis Prompt

System. You are a query analysis assistant. You must output valid JSON format.

Optional Temporal Context. Conversation context: This conversation took place between {CONV_DATE_START} and {CONV_DATE_END}. If the query uses relative or implicit time references, resolve them using this date window. A temporal signal is confirmed in this query. Extract start and end dates for any time period mentioned.

User. Analyze the following query and extract key information.

Query. {query}

Fields to Extract.

1. keywords: names, places, topic words, and other useful search terms.
2. persons: person names mentioned.
3. time_expression: time expression, if any.

4. `time_range_start`: ISO start date if a time period is mentioned.
5. `time_range_end`: ISO end date if a time period is mentioned.
6. `location`: location, if any.
7. `entities`: companies, products, organizations, or other named entities.

Output Format.

```
{
"keywords": ["keyword1", "keyword2"],
"persons": ["name1", "name2"],
"time_expression": "time expression or null",
"time_range_start": "YYYY-MM-DD or null",
"time_range_end": "YYYY-MM-DD or null",
"location": "location or null",
"entities": ["entity1"]
}
```

Constraint. Return only JSON.

Retrieval Planning Prompt

System. You are an intelligent information requirement analyst. You must output valid JSON format.

User. Analyze the following question and determine what information is required to answer it.

Question. {query}

Consider.

1. What type of question is this?
2. What entities, events, or concepts need to be identified?
3. What relationships need to be established?
4. What minimal information would be sufficient to answer the question?

Output Format.

```
{
"question_type": "factual / temporal / relational /
explanatory / other",
"key_entities": ["entity1", "entity2"],
"required_info": [
{
"info_type": "type of information",
"description": "specific information needed",
"priority": "high / medium / low"
}
],
"relationships": ["relationship1"],
"minimal_queries_needed": 2
}
```

Constraint. Focus on the minimal essential information, not exhaustive details. Return only JSON.

Answer Generation Prompt

System. You are a professional Q&A assistant. Extract concise answers from context. You must output valid JSON format.

Context Format.

```
[Context {i}]
Content: {entry.lossless_restatement}
Session: {entry.session_date} ->
{entry.session_end_date}
Time: {entry.timestamp}
Location: {entry.location}
```

```
Persons: {entry.persons}
Related Entities: {entry.entities}
Topic: {entry.topic}
```

User. Answer the user's question based on the provided context.

User Question. {query}

Relevant Context. {context_str}

Requirements.

1. Think through the reasoning process.
2. Provide a very concise answer.
3. Answer only from the provided context.
4. Format dates as "DD Month YYYY" when dates are needed.
5. Return valid JSON.

Output Format.

```
{
"reasoning": "Brief explanation of the reasoning.",
"answer": "Concise answer in a short phrase."
}
```

Constraint. Return only JSON.

C.2 Analysis of Main Results

Table 1 presents the full LoCoMo results across four backbone models and four question categories. RaMem achieves the strongest average F1 on every backbone. On closed-source models, RaMem improves average F1 from 39.06 to 51.66 on GPT-4o and from 43.24 to 54.23 on GPT-4.1-mini. The improvement is also consistent on open-source models, where average F1 increases from 33.45 to 44.55 on Qwen3-8B and from 17.98 to 24.65 on Qwen2.5-3B. Average BLEU follows the same overall trend across all four backbones. These gains are meaningful because SimpleMem is already a strong structured-memory baseline that uses semantic compression, multi-view indexing, and intent-aware retrieval. RaMem therefore provides a complementary benefit: it restores the contextual distinctions needed to recover the intended memory, rather than only making memories more compact or accessible.

The gains are especially clear on stronger backbones. On GPT-4o, RaMem improves average F1 by 12.60 points and average BLEU by 16.35 points over SimpleMem. It also achieves the best F1 in MultiHop, Temporal, and OpenDomain categories, while remaining competitive with the full-history LoCoMo setting on SingleHop questions. On GPT-4.1-mini, RaMem improves average F1 by 10.99 points and average BLEU by 8.24 points. The largest category-level improvements appear in OpenDomain and SingleHop questions, where F1 increases from 19.76 to 26.35 and from 51.12 to 58.14, respectively. This suggests that contextual reinstatement is not merely helping temporally phrased questions. It also helps the model distinguish the intended episode or state when multiple memories share entities or topics.

Open-source models show the same pattern. On Qwen3-8B, RaMem improves average F1 by 11.10 points and average BLEU by 8.33 points over SimpleMem. The method improves MultiHop F1 from 28.97 to 36.10 and SingleHop F1 from 46.62 to 50.59, showing that better episode-level recall benefits both compositional reasoning and direct factual lookup. On Qwen2.5-3B, RaMem improves average F1 from 17.98 to 24.65 and average BLEU from 14.89 to 19.44. Although this smaller model remains less capable overall, the consistent gain indicates that improving recall before generation remains useful even when the generator itself is weak.

The category-level results also clarify the role of temporal and episodic information. RaMem does improve the Temporal category on most backbones, but its advantage is not confined to temporal reasoning. On GPT-4o, RaMem improves MultiHop, OpenDomain, and SingleHop F1 by 11.42, 7.35, and 7.46 points, respectively. On GPT-4.1-mini, the largest gains are in OpenDomain and SingleHop questions. On Qwen3-8B and Qwen2.5-3B, RaMem again improves MultiHop and SingleHop

Table 5: Performance comparison on LongMemEval-S. Following the SimpleMem protocol, we use `gpt-4.1-mini` as the LLM judge and report accuracy across temporal, multi-session, knowledge-update, and single-session memory categories.

Category	Full-context	Mem0	LightMem	SimpleMem	RaMem
Temporal	27.06	40.60	85.71	83.46	**86.47**
Multi-Session	30.08	50.37	47.37	60.92	**65.41**
Knowledge-Update	41.03	69.23	**92.30**	79.48	83.33
Single-Session-User	47.14	87.14	**88.57**	85.71	87.14
Single-Session-Assistant	32.14	48.21	21.43	75.00	**78.57**
Single-Session-Preference	60.00	63.33	76.67	76.67	**80.00**
Average	39.57	59.81	68.67	76.87	**80.15**

Table 6: Context collapse analysis on the context-confusable subset. D@1 measures whether the top retrieved memory is a context distractor. RankGap is the rank of the ground-truth memory minus the rank of the best context distractor, so lower is better.

Backbone	Method	D@1 ↓	AvgD@5 ↓	RankGap ↓	GT R@10 ↑	F1 ↑
GPT-4.1-mini	SimpleMem	0.4366	3.4514	5.4135	0.6842	0.5107
	Ours	**0.3495**	**3.1336**	**2.1502**	**0.7510**	**0.5290**
Qwen3-8B	SimpleMem	0.4588	3.3909	6.1638	0.6036	0.4117
	Ours	**0.3846**	**3.0589**	**2.4664**	**0.6826**	**0.4489**
Qwen2.5-3B	SimpleMem	0.5414	3.6317	6.4248	0.5676	0.2207
	Ours	**0.4796**	**3.3554**	**3.5679**	**0.6246**	**0.2281**

performance by clear margins. These results support the main design motivation: temporal and session information is not used only to answer temporal questions. It provides contextual coordinates that help distinguish which episode, state, or event instance should be recalled across different types of memory queries.

There are also a few informative exceptions. For example, GPT-4.1-mini and Qwen3-8B show lower Temporal BLEU than SimpleMem, even though Temporal F1 improves. On Qwen2.5-3B, RaMem improves Temporal F1 over SimpleMem but has lower Temporal BLEU than SimpleMem and A-Mem. This suggests that even after the appropriate memory is recovered, answer realization can still vary in wording or granularity, especially for smaller generators and temporal answers. The overall pattern nevertheless remains consistent: RaMem improves average answer quality and achieves broad gains across categories, indicating that contextual reinstatement provides a robust complement to existing memory compression and retrieval strategies.

Table 5 further evaluates RaMem on LongMemEval-S, which stresses long-horizon memory under fine-grained temporal, update, and session-specific queries. RaMem achieves the best average accuracy, improving over SimpleMem from 76.87 to 80.15. The gains are strongest in categories that require distinguishing the correct memory episode or state, including Multi-Session, Single-Session-Assistant, and Single-Session-Preference. RaMem also slightly improves Temporal accuracy over the strongest prior result, while remaining competitive in Single-Session-User and Knowledge-Update categories where LightMem is especially strong. These results complement the LoCoMo findings: contextual reinstatement is useful not only for conversational QA with F1 or BLEU evaluation, but also under judge-based accuracy evaluation on a separate long-memory benchmark.

C.3 Context Collapse Analysis

We provide the complete context collapse analysis in Table 7. The analysis is conducted under two settings. The first setting uses all questions with verified ground-truth memories. The second focuses on the context-confusable subset, where the SimpleMem baseline retrieves at least one strict context distractor in its top-5 results. A strict context distractor is defined as a retrieved memory that is not the ground-truth memory, is content-related to the query or ground-truth memory according to person, entity, topic, or keyword overlap, and is contextually incompatible according to temporal or session mismatch. This rule-based detector does not use embedding similarity in the reported results, making the analysis conservative and interpretable.

Table 7: Full context collapse analysis. The baseline is SimpleMem. D@K measures whether at least one context distractor appears in the top-K retrieved memories. AvgDTop5 is the average number of context distractors in the top-5 memories. RankGap is the rank of the ground-truth memory minus the rank of the best context distractor. Lower values are better for D@K, AvgDTop5, and RankGap.

Setting	Backbone	Method	N	D@1 ↓	D@3 ↓	D@5 ↓	AvgDTop5 ↓	RankGap ↓	GT R@5 ↑	GT R@10 ↑	MRR ↑	F1 ↑
All Verified GT	Qwen3-8B	SimpleMem	1499	0.4416	0.8599	0.9626	3.2642	5.6976	0.5243	0.6144	0.3991	0.4138
		Ours	1499	**0.3702**	**0.8025**	**0.9320**	**2.9533**	**2.1427**	**0.6011**	**0.6925**	**0.4502**	**0.4522**
	Qwen2.5-3B	SimpleMem	1503	0.5223	0.8862	0.9647	3.5037	6.0253	0.4970	0.5768	0.3773	0.2250
		Ours	1502	**0.4627**	**0.8329**	**0.9427**	**3.2550**	**3.3185**	**0.5473**	**0.6332**	**0.4112**	**0.2315**
	GPT-4.1-mini	SimpleMem	1514	0.4273	0.8943	0.9789	3.3785	5.1934	0.6156	0.6896	0.4751	0.5109
		Ours	1514	**0.3428**	**0.8309**	**0.9551**	**3.0773**	**2.0278**	**0.6737**	**0.7550**	**0.5323**	**0.5291**
Context-Confusable	Qwen3-8B	SimpleMem	1443	0.4588	0.8933	1.0000	3.3909	6.1638	0.5100	0.6036	0.3926	0.4117
		Ours	1443	**0.3846**	**0.8337**	**0.9605**	**3.0589**	**2.4664**	**0.5925**	**0.6826**	**0.4452**	**0.4489**
	Qwen2.5-3B	SimpleMem	1450	0.5414	0.9186	1.0000	3.6317	6.4248	0.4862	0.5676	0.3735	0.2207
		Ours	1449	**0.4796**	**0.8585**	**0.9655**	**3.3554**	**3.5679**	**0.5411**	**0.6246**	**0.4094**	**0.2281**
	GPT-4.1-mini	SimpleMem	1482	0.4366	0.9136	1.0000	3.4514	5.4135	0.6093	0.6842	0.4740	0.5107
		Ours	1482	**0.3495**	**0.8468**	**0.9669**	**3.1336**	**2.1502**	**0.6694**	**0.7510**	**0.5330**	**0.5290**

Across all verified questions, contextual reinstatement consistently reduces wrong-episode interference while improving ground-truth retrieval. For GPT-4.1-mini, D@1 decreases from 0.4273 to 0.3428, AvgDTop5 decreases from 3.3785 to 3.0773, and RankGap decreases from 5.1934 to 2.0278. At the same time, GT R@10 increases from 0.6896 to 0.7550 and MRR increases from 0.4751 to 0.5323. The same pattern holds for Qwen3-8B, where D@1 decreases from 0.4416 to 0.3702, RankGap decreases from 5.6976 to 2.1427, and GT R@10 increases from 0.6144 to 0.6925. Qwen2.5-3B follows the same trend, although the final F1 gain is smaller, suggesting that weaker generators remain limited in their ability to exploit improved recall. Overall, these results show that contextual reinstatement changes the composition and ordering of recalled memories, rather than only affecting final generation behavior.

The context-confusable subset provides a more targeted stress test of context collapse. By construction, SimpleMem retrieves at least one strict context distractor in the top-5 results for every question in this subset, so D@5 is 1.0000 for the baseline. The key question is whether RaMem can reduce the dominance of these wrong-episode memories and recover the target episode more effectively. The answer is consistently positive. On GPT-4.1-mini, D@1 decreases from 0.4366 to 0.3495, RankGap decreases from 5.4135 to 2.1502, and GT R@10 increases from 0.6842 to 0.7510. On Qwen3-8B, D@1 decreases from 0.4588 to 0.3846 and RankGap decreases from 6.1638 to 2.4664, while GT R@10 increases from 0.6036 to 0.6826. On Qwen2.5-3B, D@1 decreases from 0.5414 to 0.4796 and RankGap decreases from 6.4248 to 3.5679. These results support the interpretation that contextual reinstatement is most beneficial when semantic similarity is actively misleading: it reduces the probability that a contextually incompatible memory dominates recall and makes the ground-truth memory more competitive under a fixed retrieval budget.

C.4 Content–Context Binding Results

We provide complete content–context binding results in Tables 8 and 9. The goal of this experiment is to test whether contextual reinstatement depends on a meaningful binding between memory content and its episodic context. All shuffle variants keep the memory content unchanged, including the lossless restatement and its dense embedding. Only the context fields are shuffled across memory entries within the same memory bank. Thus, the interventions preserve the same amount and format of contextual information while selectively breaking the correspondence between each memory and its original episodic coordinates.

The all-question results show that disrupting temporal/session binding harms both recall and answer quality across different backbones. On GPT-4.1-mini, temporal shuffling reduces GT R@10 from 0.7550 to 0.6744 and increases D@1 from 0.3428 to 0.5753. RankGap also increases from 2.0278 to 5.2974, meaning that contextually incompatible memories again dominate the ground-truth memory in the ranking. Qwen3-8B shows the same trend: GT R@10 drops from 0.6925 to 0.6204, D@1 increases from 0.3702 to 0.6518, and RankGap rises from 2.1427 to 5.7768. Full-context shuffling produces similar degradation on both backbones. These results indicate that correct temporal/session binding shapes recall across the benchmark, rather than only within a narrow subset of examples.

Table 8: Context shuffle test on all verified ground-truth questions. Lower is better for CappedRank@10, D@1, AvgDTop5, and RankGap.

Backbone	Variant	N	F1 ↑	BLEU1 ↑	GT R@5 ↑	GT R@10 ↑	MRR ↑	CappedRank@10 ↓	D@1 ↓	AvgDTop5 ↓	RankGap ↓
GPT-4.1-mini	Original	1514	0.5291	0.4475	0.6737	0.7550	0.5323	5.5132	0.3428	3.0773	2.0278
	Shuffle temporal	1514	0.5004	0.4230	0.6103	0.6744	0.4871	7.6149	0.5753	4.0396	5.2974
	Shuffle entity	1514	0.5256	0.4444	0.6830	0.7523	0.5349	5.4465	0.3382	3.0918	1.9761
	Shuffle topic	1514	0.5255	0.4441	0.6777	0.7523	0.5327	5.4815	0.3408	3.0746	1.9861
	Shuffle full context	1514	0.5079	0.4296	0.6050	0.6638	0.4822	7.6876	0.5872	4.0707	5.3965
Qwen3-8B	Original	1499	0.4522	0.3801	0.6011	0.6925	0.4502	6.2028	0.3702	2.9533	2.1427
	Shuffle temporal	1499	0.4287	0.3606	0.5484	0.6204	0.4116	8.1134	0.6518	4.0747	5.7768
	Shuffle entity	1499	0.4526	0.3801	0.5997	0.6898	0.4517	6.2141	0.3689	3.0013	2.1699
	Shuffle topic	1499	0.4546	0.3828	0.6051	0.6898	0.4519	6.1654	0.3722	2.9533	2.0862
	Shuffle full context	1499	0.4347	0.3642	0.5437	0.6157	0.4053	8.1161	0.6631	4.1361	5.7634

Table 9: Context shuffle test on temporal-triggered questions. This subset isolates questions where temporal/session context is explicitly activated. Lower is better for CappedRank@10, D@1, AvgDTop5, and RankGap.

Backbone	Variant	N	F1 ↑	BLEU1 ↑	GT R@5 ↑	GT R@10 ↑	MRR ↑	CappedRank@10 ↓	D@1 ↓	AvgDTop5 ↓	RankGap ↓
GPT-4.1-mini	Original	212	0.6009	0.5396	0.6934	0.8066	0.5249	5.3113	0.2642	2.1462	1.0404
	Shuffle temporal	212	0.4985	0.4396	0.1981	0.2358	0.1796	21.0802	0.8160	4.2972	21.5029
	Shuffle entity	212	0.5843	0.5198	0.6887	0.7972	0.5184	5.4198	0.2830	2.2453	1.3266
	Shuffle topic	212	0.5982	0.5309	0.6887	0.8019	0.5171	5.4057	0.2736	2.1462	1.2211
	Shuffle full context	212	0.5180	0.4578	0.1887	0.2358	0.1762	21.3915	0.8302	4.3915	22.0116
Qwen3-8B	Original	212	0.5317	0.4810	0.5708	0.7170	0.4271	6.6038	0.2736	2.0142	1.2350
	Shuffle temporal	212	0.4081	0.3681	0.1651	0.1840	0.1250	20.3255	0.8726	4.3443	22.9595
	Shuffle entity	212	0.5075	0.4596	0.5613	0.7170	0.4238	6.5142	0.2736	2.0802	1.1585
	Shuffle topic	212	0.5195	0.4677	0.5660	0.7123	0.4286	6.4528	0.2783	2.0000	1.0546
	Shuffle full context	212	0.4134	0.3694	0.1557	0.1840	0.1242	20.3491	0.8868	4.5094	23.0952

The temporal-triggered subset provides a sharper mechanism-level view. On GPT-4.1-mini, shuffling temporal fields causes GT R@10 to collapse from 0.8066 to 0.2358 and MRR from 0.5249 to 0.1796. At the same time, D@1 increases from 0.2642 to 0.8160 and RankGap rises from 1.0404 to 21.5029. Qwen3-8B exhibits the same behavior: GT R@10 drops from 0.7170 to 0.1840, MRR from 0.4271 to 0.1250, D@1 increases from 0.2736 to 0.8726, and RankGap rises from 1.2350 to 22.9595. Full-context shuffling closely matches this failure pattern, suggesting that the main source of degradation is the disrupted temporal/session binding. By contrast, shuffling entity or topic fields has much smaller effects. This finding clarifies the current implementation: contextual reinstatement is primarily operationalized through session-level temporal grounding, while entity and topic fields serve as auxiliary cues. Overall, the results show that contextual information is useful only when it remains correctly bound to memory content, and that breaking this binding reintroduces the context collapse that RaMem is designed to mitigate.

C.5 Temporal Buffer Sensitivity

Several consistent trends emerge. First, the best performance is obtained with a *moderate* temporal buffer rather than with the strictest or widest setting. On GPT-4.1-mini, the 5-day window gives the strongest overall performance on all verified questions, achieving an F1 of 0.5291 and a GT R@10 of 0.7550, while 3–5 days form a similarly strong region on the temporal-triggered subset. On Qwen3-8B, the optimum shifts slightly toward a narrower setting, with 3 days producing the strongest temporal-triggered performance and near-best overall performance. This pattern is informative: the system benefits from some tolerance to event-time and mention-time mismatch, but the optimal tolerance is not unbounded. The results therefore support the design intuition of contextual reinstatement. Temporal context should function as a calibrated episodic coordinate, rather than as a hard equality constraint.

Second, enlarging the buffer too aggressively weakens episodic specificity and reintroduces competing memories from nearby but incompatible contexts. This is most visible at 14 days. On GPT-4.1-mini, all-verified GT R@10 drops from 0.7550 at 5 days to 0.7417 at 14 days, while on the temporal-triggered subset RankGap increases from 1.0404 to 2.5736 and AvgDTop5 rises from 2.1462 to 2.3821. Qwen3-8B exhibits an even clearer degradation: on temporal-triggered questions, F1 drops from 0.5329 at 3 days to 0.4799 at 14 days, while GT R@10 decreases from 0.7311 to 0.6651. These changes show that a broader temporal window does not monotonically improve recall. Instead, it eventually admits more contextually incompatible memories and weakens the ranking advantage of the target episode. Taken together, the two backbones tell a consistent story: contextual reinstatement

Table 10: Temporal buffer sensitivity on all verified questions. Lower is better for D@1, RankGap, and AvgDTop5.

Backbone	Buffer	N	F1 ↑	GT R@10 ↑	MRR ↑	D@1 ↓	RankGap ↓	AvgDTop5 ↓
GPT-4.1-mini	0	1513	0.5226	0.7396	0.5279	0.3410	1.6488	3.0330
	1	1514	0.5172	0.7490	0.5304	0.3461	1.5294	3.0297
	3	1514	0.5270	0.7483	0.5321	0.3435	1.7790	3.0396
	5	1514	**0.5291**	**0.7550**	**0.5323**	0.3428	2.0278	3.0773
	7	1514	0.5245	0.7503	0.5340	**0.3415**	1.9992	3.0727
	14	1514	0.5224	0.7417	0.5302	0.3428	2.1756	3.0991
Qwen3-8B	0	1499	0.4496	0.6905	0.4521	0.3682	1.6099	2.9333
	1	1499	**0.4558**	0.6918	0.4523	**0.3649**	1.5397	**2.9260**
	3	1499	0.4557	**0.6938**	**0.4528**	**0.3649**	1.7316	2.9386
	5	1499	0.4522	0.6925	0.4502	0.3702	2.1427	2.9533
	7	1499	0.4532	0.6865	0.4465	0.3696	2.2101	2.9606
	14	1499	0.4499	0.6865	0.4485	0.3769	2.2721	2.9960

Table 11: Temporal buffer sensitivity on the temporal-triggered subset. Lower is better for D@1, RankGap, and AvgDTop5.

Backbone	Buffer	N	F1 ↑	GT R@10 ↑	MRR ↑	D@1 ↓	RankGap ↓	AvgDTop5 ↓
GPT-4.1-mini	0	212	0.5954	0.7358	0.4837	0.2783	-0.5663	1.9481
	1	212	0.5841	0.7877	0.5196	**0.2594**	-1.9137	**1.9198**
	3	212	**0.6146**	0.7925	0.5126	0.2689	-0.2929	2.0236
	5	212	0.6009	**0.8066**	0.5249	0.2642	1.0404	2.1462
	7	212	0.5931	0.7925	**0.5250**	0.2830	1.7828	2.2123
	14	212	0.5906	0.7547	0.5088	0.2925	2.5736	2.3821
Qwen3-8B	0	212	0.4987	0.6934	0.4145	0.2736	-1.5543	1.8679
	1	212	0.5187	0.7217	0.4293	0.2594	-2.3118	**1.8019**
	3	212	**0.5329**	**0.7311**	**0.4332**	**0.2547**	-0.6237	1.8632
	5	212	0.5317	0.7170	0.4271	0.2736	1.2350	2.0142
	7	212	0.5231	0.7028	0.4235	0.2689	1.7611	2.0708
	14	212	0.4799	0.6651	0.4066	0.3066	2.9257	2.3443

requires a *moderate* temporal window that absorbs small temporal mismatches while preserving episodic specificity.

We provide the full temporal buffer sensitivity results in Tables 10 and 11. The goal of this experiment is to understand how the temporal reinstatement window controls the balance between recall tolerance and contextual specificity. A zero-day window is maximally strict and only admits temporally aligned memories, while larger windows allow increasing tolerance for mismatch between event time and mention time. In principle, a larger buffer can help recover memories that belong to the target episode but were mentioned slightly outside the exact query span. However, an excessively broad buffer may also admit nearby memories from incompatible episodic contexts, thereby increasing distractor exposure. We therefore evaluate buffer sizes of 0, 1, 3, 5, 7, and 14 days on both GPT-4.1-mini and Qwen3-8B.

The results show that the temporal reinstatement window should be neither maximally strict nor excessively broad. On all verified questions, both backbones achieve their strongest or near-strongest performance in the moderate range between 3 and 5 days. GPT-4.1-mini performs best at 5 days, where F1 reaches 0.5291 and GT R@10 reaches 0.7550. Qwen3-8B is slightly more conservative, with 1–3 days giving the strongest overall results and 3 days producing the best GT R@10. This pattern is consistent with the intended role of the buffer. Some tolerance is needed to absorb mild mismatch between event time and mention time, but once the window becomes too large, contextual reinstatement loses the specificity needed to separate the target episode from nearby alternatives.

The trade-off is much clearer on the temporal-triggered subset, where temporal reinstatement directly constrains recall. Here, moderate windows again dominate. GPT-4.1-mini achieves its highest F1 at 3 days and its highest GT R@10 at 5 days, while Qwen3-8B peaks at 3 days on F1, GT R@10, MRR, and D@1. In contrast, the 14-day setting consistently degrades performance. For GPT-4.1-mini,

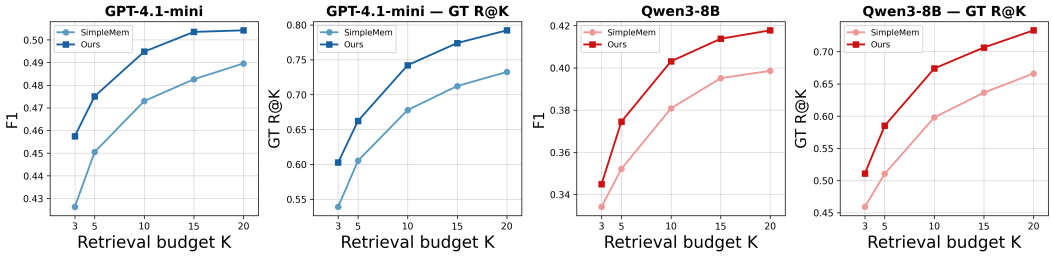


Figure 5: Sensitivity to retrieval budget. Each subplot varies the number of retrieved memories passed to the generator and compares SimpleMem with our method. Across both backbones, contextual reinstatement consistently improves F1 and GT R@K under all budgets, with particularly clear advantages in the low-budget regime.

temporal-triggered RankGap increases from 1.0404 at 5 days to 2.5736 at 14 days, and AvgDTop5 increases from 2.1462 to 2.3821. For Qwen3-8B, F1 drops from 0.5329 at 3 days to 0.4799 at 14 days, while GT R@10 decreases from 0.7311 to 0.6651 and D@1 rises from 0.2547 to 0.3066. These results support a clear interpretation: a moderate temporal window offers the best balance between tolerating temporal mismatch and excluding contextually incompatible memories. In other words, contextual reinstatement benefits from calibrated flexibility, not from indiscriminately widening the admissible recall region.

C.6 Retrieval Budget Sensitivity

To evaluate whether contextual reinstatement remains effective under limited context budgets, we vary the number of retrieved memories passed to the generator and compare our method with the SimpleMem baseline. This experiment is important because the main claim of our framework is not merely that it retrieves more memories, but that it makes the target episode accessible earlier in the ranked list. If this claim is correct, the advantage should already appear under small budgets, where the generator can observe only a few retrieved entries. Figure 5 reports the trends on GPT-4.1-mini and Qwen3-8B, using F1 and budget-aligned ground-truth recall (GT R@K) to capture both final answer quality and retrieval effectiveness. Two clear patterns emerge. First, contextual reinstatement consistently outperforms the baseline across the entire budget range. On GPT-4.1-mini, our method improves F1 from 0.4575 to 0.5042 as K increases from 3 to 20, while remaining above the baseline at every budget. The same trend holds for budget-aligned ground-truth recall, where our method improves from 0.6026 to 0.7922 compared with the baseline range of 0.5390 to 0.7325. Qwen3-8B exhibits the same behavior: our method dominates the baseline on both F1 and GT R@K for every retrieval budget. This result supports the main intuition of contextual reinstatement. The gain does not depend on retrieving a large number of memories. Instead, it appears already at small budgets, showing that the method improves recall ordering and makes the answer-supporting episode accessible earlier in the ranked list.

Second, the low-budget regime is especially informative. Under tight budgets, the generator must rely on only a few retrieved memories, so recall ordering matters most. Here the advantage of our method is particularly meaningful. On GPT-4.1-mini, our method at $K = 10$ already exceeds the baseline at $K = 20$ on both F1 (0.4948 vs. 0.4896) and GT R@K (0.7422 vs. 0.7325). The same pattern appears on Qwen3-8B, where our method at $K = 10$ surpasses the baseline at $K = 20$ on both F1 (0.4031 vs. 0.3986) and GT R@K (0.6740 vs. 0.6662). This shows that contextual reinstatement improves not only answer quality but also *context efficiency*: it can reach the performance of a larger retrieval budget while using fewer retrieved memories. In practical agent systems, where context budget is often constrained, this behavior is especially valuable.

We provide the full retrieval-budget sensitivity results in Table 12. The purpose of this experiment is to test whether contextual reinstatement merely increases the chance that the target memory appears somewhere in a large retrieved set, or instead improves its position under constrained budgets. We vary the number of retrieved memories passed to the generator from $K = 3$ to $K = 20$, and report final answer quality, budget-aligned ground-truth recall (GT R@K), MRR, and two context-collapse diagnostics. Because D@1 measures whether the top-ranked memory is a context distractor, it is

Table 12: Sensitivity to retrieval budget. GT R@K denotes the fraction of questions whose ground-truth memory is included within the top-K retrieved memories. Lower is better for D@1 and RankGap.

Backbone	Method / K	F1 ↑	BLEU1 ↑	GT R@K ↑	MRR ↑	D@1 ↓	RankGap ↓
GPT-4.1-mini	SimpleMem, $K = 3$	0.4263	0.3530	0.5390	0.4344	0.4201	-0.5637
	SimpleMem, $K = 5$	0.4505	0.3739	0.6052	0.4496	0.4201	-0.4452
	SimpleMem, $K = 10$	0.4730	0.3942	0.6779	0.4593	0.4201	0.1772
	SimpleMem, $K = 15$	0.4826	0.4033	0.7123	0.4620	0.4201	0.7028
	SimpleMem, $K = 20$	0.4896	0.4107	0.7325	0.4632	0.4201	1.1392
	Ours, $K = 3$	0.4575	0.3793	0.6026	0.4932	0.3370	-0.7108
	Ours, $K = 5$	0.4751	0.3985	0.6623	0.5069	0.3370	-0.7696
	Ours, $K = 10$	0.4948	0.4167	0.7422	0.5180	0.3370	-0.2264
	Ours, $K = 15$	0.5035	0.4244	0.7740	0.5205	0.3370	0.1835
	Ours, $K = 20$	0.5042	0.4228	0.7922	0.5215	0.3370	0.5321
Qwen3-8B	SimpleMem, $K = 3$	0.3341	0.2721	0.4591	0.3564	0.4299	-0.4991
	SimpleMem, $K = 5$	0.3521	0.2881	0.5104	0.3679	0.4299	-0.5054
	SimpleMem, $K = 10$	0.3808	0.3144	0.5981	0.3798	0.4299	0.2013
	SimpleMem, $K = 15$	0.3951	0.3269	0.6364	0.3828	0.4299	0.8306
	SimpleMem, $K = 20$	0.3986	0.3311	0.6662	0.3845	0.4299	1.5361
	Ours, $K = 3$	0.3449	0.2822	0.5110	0.4032	0.3604	-0.6649
	Ours, $K = 5$	0.3745	0.3107	0.5851	0.4203	0.3604	-0.6400
	Ours, $K = 10$	0.4031	0.3329	0.6740	0.4323	0.3604	-0.1169
	Ours, $K = 15$	0.4138	0.3422	0.7065	0.4350	0.3604	0.2999
	Ours, $K = 20$	0.4178	0.3477	0.7331	0.4364	0.3604	0.8714

unaffected by the truncation budget and remains constant for a fixed retrieval system. RankGap is also reported for completeness, although it is most informative when comparing methods at the same budget rather than when comparing different budgets.

The results show that contextual reinstatement improves performance under every tested retrieval budget. On GPT-4.1-mini, the baseline F1 rises from 0.4263 at $K = 3$ to 0.4896 at $K = 20$, while our method rises from 0.4575 to 0.5042. The same consistent dominance appears in GT R@K, where our method improves from 0.6026 to 0.7922 compared with the baseline range of 0.5390 to 0.7325. Qwen3-8B follows the same pattern: our method improves F1 from 0.3449 to 0.4178 and GT R@K from 0.5110 to 0.7331, remaining above the baseline at every budget. MRR is also consistently higher for our method on both backbones, indicating that the answer-supporting memory is ranked closer to the top of the recalled set. D@1 remains lower for our method on both backbones, showing that the top-ranked memory is less likely to come from an incompatible episode. Together, these results confirm that the gains are not tied to a specific budget size.

More importantly, the experiment reveals a clear efficiency advantage. On GPT-4.1-mini, our method with $K = 10$ already exceeds the baseline with $K = 20$ on both F1 (0.4948 vs. 0.4896) and GT R@K (0.7422 vs. 0.7325). The same effect holds on Qwen3-8B, where our method at $K = 10$ surpasses the baseline at $K = 20$ on F1 (0.4031 vs. 0.3986) and GT R@K (0.6740 vs. 0.6662). This means that contextual reinstatement does more than improve final accuracy: it allows the system to reach the same or better performance using fewer retrieved memories. Such behavior is particularly valuable in practical long-context agent systems, where retrieval budget is tightly coupled to latency and prompt cost. The results therefore support a stronger claim than backbone-level improvement alone: contextual reinstatement yields a better performance–budget trade-off by improving recall ordering and reducing wrong-episode interference under constrained context.

C.7 Efficiency and Cost Results

We provide additional efficiency and cost results in Tables 14, 15, and 16. All runtime numbers compare SimpleMem-style retrieval and our method under the same optimized implementation. They should not be interpreted as a direct comparison against the original SimpleMem codebase, since our system uses a different serving stack with batching, parallel execution, and other engineering optimizations. The purpose of this analysis is to estimate the incremental cost of contextual reinstatement under a shared runtime environment.

Table 13: Context-budget efficiency. RaMem with $K = 10$ reaches or exceeds SimpleMem with $K = 20$ while using about half the retrieved memory budget.

Backbone	Method	K	F1 ↑	GT R@K ↑	CtxTok/Q ↓
GPT-4.1-mini	SimpleMem	20	0.4896	0.7325	1520.71
	RaMem	10	**0.4948**	**0.7422**	**758.24**
Qwen3-8B	SimpleMem	20	0.3986	0.6662	1367.86
	RaMem	10	**0.4031**	**0.6740**	**683.50**

Table 14: Default online cost under the same optimized serving stack. Generation time and total time are unavailable in the historical full-evaluation logs and are therefore marked as NA.

Backbone	Method	F1 ↑	GT R@10 ↑	LLM Calls/Q ↓	Retrieval Time/Q ↓	Generation Time/Q ↓	Total Time/Q ↓	ContextChars/Q ↓	Retrieved Memories/Q ↓
GPT-4.1-mini	SimpleMem	0.5104	0.6779	4.0000	4.8346s	NA	NA	14692.86	48.59
	Ours	**0.5281**	**0.7422**	4.0000	4.9076s	NA	NA	**13261.07**	**43.86**
Qwen3-8B	SimpleMem	0.4105	0.5981	4.0000	7.2318s	NA	NA	12715.18	46.72
	Ours	**0.4459**	**0.6740**	4.0000	**6.6148s**	NA	NA	**11399.62**	**41.91**

The default-cost results show that contextual reinstatement does not require additional LLM calls under our planning-on and reflection-off setting. Both SimpleMem-style retrieval and our method use approximately four LLM calls per question. The added contextual operations are mainly structured filtering, temporal/session compatibility checking, and reranking, which are lightweight compared with LLM generation. On GPT-4.1-mini, retrieval time changes only marginally, while context length and retrieved memory count decrease. On Qwen3-8B, retrieval time also decreases under the same optimized stack. Since generation and total wall-clock time were not stored in the full historical logs, those columns are marked as NA in Table 14.

The budget-efficiency results provide a more robust view of cost effectiveness. On both backbones, our method at $K = 10$ reaches or exceeds SimpleMem at $K = 20$ while using approximately half the retrieved memory budget. For GPT-4.1-mini, our method at $K = 10$ achieves F1 of 0.4948 and GT R@K of 0.7422, compared with 0.4896 and 0.7325 for SimpleMem at $K = 20$. For Qwen3-8B, our method at $K = 10$ achieves F1 of 0.4031 and GT R@K of 0.6740, compared with 0.3986 and 0.6662 for SimpleMem at $K = 20$. This indicates that contextual reinstatement improves recall ordering and makes the target memory accessible under a smaller context budget, allowing the generator to receive fewer but more informative memory entries.

Finally, the small-sample timing sanity check in Table 16 is consistent with the default-cost analysis. Under the same optimized stack, our method does not increase total online time and often reduces it slightly because fewer memories are passed to the generator. However, these timing numbers are based on a small sample and should be interpreted as supporting evidence rather than as a standalone runtime benchmark. The main efficiency conclusion is therefore based on controlled within-stack cost comparison and context-budget efficiency, rather than direct runtime comparison to external codebases.

C.8 Full Mechanism Analysis

We provide the complete ablation results in Tables 17 and 18. The main text reports the most diagnostic ablations on the temporal-triggered subset, while this appendix includes both all verified questions and temporal-triggered questions, as well as the cue-guard ablation. The cue guard disables a deterministic reliability check for contextual activation. In our experiments, this ablation has relatively small effects compared with removing session-level contextual coordinates or context-aware ranking, so we treat it as a reliability mechanism rather than the primary source of the observed improvements.

The complete results confirm the pattern discussed in the main text. Removing session context or context-aware ranking consistently harms recall quality. On all verified questions, these ablations reduce GT R@10 and MRR while increasing D@1, AvgDTop5, and RankGap. The effect is substantially stronger on temporal-triggered questions, where session-level contextual coordinates are directly used to distinguish the target episode from competing memories. For example, on Qwen3-8B, removing session context reduces GT R@10 from 0.7217 to 0.4245, while removing context-aware

Table 15: Budget efficiency under different retrieval budgets. GT R@K denotes the fraction of questions whose ground-truth memory is included within the top-K retrieved memories.

Backbone	Method	K	F1 ↑	GT R@K ↑	MRR ↑	ContextTokens/Q ↓	Retrieved Memories/Q ↓
GPT-4.1-mini	Ours	3	0.4575	0.6026	0.4932	226.48	3
	Ours	5	0.4751	0.6623	0.5069	377.29	5
	Ours	10	0.4948	0.7422	0.5180	758.24	10
	SimpleMem	10	0.4730	0.6779	0.4593	759.49	10
	SimpleMem	20	0.4896	0.7325	0.4632	1520.71	20
Qwen3-8B	Ours	10	0.4031	0.6740	0.4323	683.50	10
	SimpleMem	10	0.3808	0.5981	0.3798	685.16	10
	SimpleMem	20	0.3986	0.6662	0.3845	1367.86	20

Table 16: Small-sample timing sanity check under the optimized serving stack. These results are included for transparency and should be interpreted as within-stack timing estimates rather than definitive throughput benchmarks.

Backbone	Method	F1 ↑	GT R@10 ↑	LLM Calls/Q	Retrieval Time/Q ↓	Generation Time/Q ↓	Total Time/Q ↓	Retrieved Memories/Q ↓	N
GPT-4.1-mini	SimpleMem	0.5104	0.6779	4.0000	5.3867s	1.6344s	7.0210s	50.81	16
	Ours	**0.5281**	**0.7422**	4.0000	**5.2808s**	**1.5286s**	**6.8093s**	**44.25**	16
Qwen3-8B	SimpleMem	0.4105	0.5981	4.0000	3.5873s	1.3234s	4.9107s	51.88	16
	Ours	**0.4459**	**0.6740**	4.0000	**3.4441s**	**1.1860s**	**4.6302s**	**44.31**	16

ranking reduces it to 0.4434. This shows that session grounding and validity-constrained ordering are the two most important recall-side mechanisms.

The ablation of context-preserved generation behaves differently. It leaves retrieval metrics unchanged, because the retrieved memories are the same, but reduces final answer quality. This confirms that recovering the appropriate episode is not sufficient if its contextual identity is discarded before downstream reasoning; the content–context binding must remain available to the generator. The cue-guard ablation has smaller and less consistent effects. In some settings, final F1 is slightly higher without the guard, but retrieval diagnostics are generally similar or mildly worse. This suggests that the guard mainly serves as a lightweight reliability mechanism for avoiding unsupported contextual constraints, while the main gains come from session-level contextual reinstatement, validity-constrained recall, and preservation of episodic binding.

C.9 Efficiency and Cost Analysis

We further analyze whether contextual reinstatement improves long-term recall at the cost of additional online overhead. Since our implementation uses an optimized serving stack with vLLM-style batching, parallel evaluation, and other engineering optimizations, we do not compare wall-clock runtime against the original SimpleMem codebase. Instead, we compare SimpleMem-style retrieval and our method under the same optimized implementation. This provides a fair estimate of the incremental cost introduced by contextual reinstatement. As shown in Table 19, both methods use the same number of LLM calls per question under the planning-on and reflection-off setting. Contextual reinstatement does not introduce iterative reflection or extra generation calls. On GPT-4.1-mini, retrieval time changes only marginally from 4.83s to 4.91s per question, while F1 improves from 0.5104 to 0.5281 and GT R@10 improves from 0.6779 to 0.7422. On Qwen3-8B, our method is slightly faster in the same runtime stack, with retrieval time decreasing from 7.23s to 6.61s, while F1 improves from 0.4105 to 0.4459. The retrieved context is also smaller in both settings, reducing average context characters from 14.7K to 13.3K on GPT-4.1-mini and from 12.7K to 11.4K on Qwen3-8B.

The benefit is even clearer when viewed as context-budget efficiency. Table 13 compares our method with $K = 10$ retrieved memories against SimpleMem with $K = 20$. On GPT-4.1-mini, our method with half the retrieved memory budget achieves higher F1 (0.4948 vs. 0.4896) and higher GT recall (0.7422 vs. 0.7325), while using roughly half the estimated context tokens. Qwen3-8B shows the same pattern: our method at $K = 10$ exceeds SimpleMem at $K = 20$ in both F1 (0.4031 vs. 0.3986) and GT recall (0.6740 vs. 0.6662), again with about half the context tokens. These results show that contextual reinstatement improves recall ordering and makes the target memory accessible under a smaller context budget, rather than relying on larger prompts or additional LLM reasoning. In

Table 17: Full ablation results on GPT-4.1-mini. Lower is better for D@1, AvgDTop5, and RankGap.

Subset	Variant	N	F1 ↑	BLEU1 ↑	GT R@10 ↑	MRR ↑	D@1 ↓	AvgDTop5 ↓	RankGap ↓
All Verified GT	Full	1514	**0.5298**	**0.4473**	**0.7483**	**0.5335**	**0.3435**	**3.0581**	**2.7454**
	w/o Session Context	1514	0.5253	0.4442	0.6909	0.4857	0.4069	3.3963	5.5944
	w/o Cue Guard	1514	0.5289	0.4472	0.7351	0.5191	0.3567	3.0859	3.2846
	w/o Context-Aware Ranking	1514	0.5262	0.4459	0.6915	0.4781	0.4128	3.3943	5.9129
	w/o Context-Preserved Generation	1514	0.5133	0.4346	**0.7483**	**0.5335**	**0.3435**	**3.0581**	**2.7454**
Temporal-triggered	Full	212	0.5957	**0.5319**	**0.8066**	0.5205	**0.2736**	2.1509	1.0354
	w/o Session Context	212	0.5596	0.4922	0.5708	0.3512	0.5189	3.6132	10.0492
	w/o Cue Guard	212	**0.5970**	**0.5319**	0.8019	0.5203	0.2783	**2.1462**	**0.9848**
	w/o Context-Aware Ranking	212	0.5693	0.5018	0.5708	0.3468	0.5283	3.5896	12.6332
	w/o Context-Preserved Generation	212	0.5626	0.4968	**0.8066**	0.5205	**0.2736**	2.1509	1.0354

Table 18: Full ablation results on Qwen3-8B. Lower is better for D@1, AvgDTop5, and RankGap.

Subset	Variant	N	F1 ↑	BLEU1 ↑	GT R@10 ↑	MRR ↑	D@1 ↓	AvgDTop5 ↓	RankGap ↓
All Verified GT	Full	1499	0.4521	0.3807	**0.6938**	**0.4519**	**0.3716**	**2.9460**	**2.0695**
	w/o Session Context	1499	0.4423	0.3705	0.6071	0.4008	0.4396	3.2602	5.8922
	w/o Cue Guard	1499	**0.4546**	**0.3831**	0.6838	0.4467	0.3769	2.9740	2.6675
	w/o Context-Aware Ranking	1499	0.4448	0.3745	0.6177	0.4062	0.4336	3.2548	6.2023
	w/o Context-Preserved Generation	1499	0.4077	0.3391	**0.6938**	**0.4519**	**0.3716**	**2.9460**	**2.0695**
Temporal-triggered	Full	212	0.5226	0.4695	**0.7217**	0.4253	0.2783	2.0142	1.1685
	w/o Session Context	212	0.4578	0.4097	0.4245	0.2482	0.5849	3.4906	11.9565
	w/o Cue Guard	212	**0.5292**	**0.4781**	0.7170	**0.4270**	**0.2736**	**2.0094**	**1.0489**
	w/o Context-Aware Ranking	212	0.4642	0.4162	0.4434	0.2578	0.5755	3.5472	14.7735
	w/o Context-Preserved Generation	212	0.4248	0.3567	**0.7217**	0.4253	0.2783	2.0142	1.1685

practical long-term agent systems, this means the agent can reach the same or better performance while passing fewer memories to the generator.

C.10 Qualitative Case Studies

We provide additional qualitative examples with the actual top retrieved memories from SimpleMem and RaMem. These cases show that context collapse is not caused by completely irrelevant retrieval. The baseline often retrieves memories that share entities or topics with the query but originate from the wrong session or interaction episode. RaMem instead promotes memories whose episodic coordinates are compatible with the context reinstated from the query, allowing the target episode to remain distinguishable from semantically related alternatives.

Case A: New Hobby on July 9

Query: What new hobby did James become interested in on 9 July, 2022?
Reference: Extreme sports.
SimpleMem answer: surfing **RaMem answer:** extreme sports.

SimpleMem Top-3:

1. Invalid: `2022-03-20--2022-03-27`, gaming platform participation. James created a game avatar and joined a new gaming platform.
2. Invalid: `2022-09-20--2022-10-03`, game streaming feedback. James streamed a game and received encouraging comments.
3. Invalid: `2022-09-18--2022-09-20`, game streaming plans. James started streaming games.

GT rank: 44.

RaMem Top-3:

1. GT / valid: `2022-07-09--2022-07-22`, extreme sports interest. James became interested in extreme sports and did rope jumping.
2. Invalid: `2022-06-19--2022-07-09`, surfing experience. James went surfing.
3. Related but topic-conflicting: `2022-07-09--2022-07-22`, gaming tournament victory. James won an online gaming tournament.

GT rank: 1.

Table 19: Online efficiency under the same optimized serving stack. We compare SimpleMem-style retrieval and our method within the same implementation, rather than against the original SimpleMem runtime. Contextual reinstatement improves answer quality and retrieval quality without increasing LLM calls.

Backbone	Method	F1 ↑	GT R@10 ↑	LLM Calls/Q ↓	Retrieval Time/Q ↓	ContextChars/Q ↓	Retrieved Memories/Q ↓
GPT-4.1-mini	SimpleMem	0.5104	0.6779	4.0000	4.8346s	14692.86	48.59
	Ours	**0.5281**	**0.7422**	4.0000	4.9076s	**13261.07**	**43.86**
Qwen3-8B	SimpleMem	0.4105	0.5981	4.0000	7.2318s	12715.18	46.72
	Ours	**0.4459**	**0.6740**	4.0000	**6.6148s**	**11399.62**	**41.91**

Observation. SimpleMem retrieves memories that are related to James and activities, but all top-ranked evidence violates the query's recall conditions. RaMem reinstates the July 9 session context and places the verified memory at rank 1, turning retrieved memory from related fragments into contextually verifiable evidence.

Case B: Vacation Episode vs. Later Partner-Related Memories

Query: When did Evan get back from a vacation with his SO?
Reference: August 13, 2023.
SimpleMem answer: The context does not specify when Evan returned.
RaMem answer: August 13, 2023.

SimpleMem Top-3:

1. Invalid: `2023-12-31--2024-01-06`, accident and marriage news. Evan confirmed that he and his friends were fine after an accident.
2. Invalid: `2024-01-06--2024-01-10`, family and partner support. Sam discussed Evan's partner and family support.
3. Invalid: `2024-01-06--2024-01-10`, family support after marriage announcement. Evan mentioned support from extended family.

GT rank: 18.

RaMem Top-3:

1. GT / valid: `2023-08-13--2023-08-15`, vacation details. Evan shared details about a recent vacation in Canada with his new partner.
2. Invalid: `2023-12-31--2024-01-06`, accident and marriage news. Evan confirmed that he and his friends were fine after an accident.
3. Invalid: `2023-11-21--2023-12-05`, mutual support. Evan thanked Sam for being there.

GT rank: 1.

Observation. The baseline retrieves partner-related memories, but they belong to later sessions. RaMem identifies the vacation episode that satisfies the query's temporal and participant conditions.

Case C: Gaming Topic vs. Correct Game Preference Episode

Query: What game has Nate been playing nonstop with a futuristic setting and gameplay on October 9, 2022?
Reference: Cyberpunk 2077.
SimpleMem answer: Catan **RaMem answer:** Cyberpunk 2077.

SimpleMem Top-3:

1. Invalid: `2022-06-03--2022-06-05`, regional video game tournament. Nate shared a recent victory in a regional video game tournament.

2. Invalid: `2022-08-22--2022-09-05`, gaming tournament and career. Nate won an international tournament and mentioned gaming as a career.
3. Invalid: `2022-04-21--2022-05-02`, gaming tournament mention. Nate mentioned an upcoming gaming tournament.

GT rank: 32.

RaMem Top-3:

1. Same session / related: `2022-10-09--2022-10-21`, board game meeting. Nate met people playing the same board game.
2. Same session / related: `2022-10-09--2022-10-21`, game convention attendance. Nate attended a game convention and met new people.
3. GT / valid: `2022-10-09--2022-10-21`, recent movie and game preferences. Nate mentioned playing *Cyberpunk 2077*.

GT rank: 3.

Observation. The baseline retrieves memories from the broad gaming topic, but they belong to earlier episodes. RaMem first narrows retrieval to the correct October 9 session, then includes the verified game-preference memory in the top retrieved evidence.