LEARNING TO RETRIEVE MISSING EVIDENCE FOR LONG-TERM MEMORY QA

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory enables language models to use past interactions in future conversations. However, evidence needed to answer a question may be scattered across distant turns, while the question itself omits clues needed to locate it. Retrieved facts can reveal these clues, motivating retrieval decisions conditioned on evidence already found. We introduce MERA (Missing-Evidence Retrieval Augmentation), which separates globally searchable memory from a question-specific evidence state. Verified evidence guides subsequent retrieval without restricting access to the global memory. We train a lightweight planner through reinforcement learning, rewarding queries that recover previously missing evidence. MERA achieves strong answer accuracy across Qwen3-30B and GPT-4o-mini backbones. With Qwen3-30B for evidence processing and answer generation, the trained 0.6B planner achieves 77.40% accuracy on LoCoMo and 71.29% on LongMemEval-S, exceeding a 30B planner without retrieval-grounded training by 4.10% and 3.96%, respectively. On LoCoMo, later retrieval rounds increase cumulative evidence recall from 55.5% to 80.5%.

1 INTRODUCTION

Long-term memory enables language models to retain information from past interactions and use it in future conversations Zhong et al. (2024); Park et al. (2023). Recent memory systems have made substantial progress in deciding what to store, how to organize it, and how to maintain it over time Zhang et al. (2025); Fang et al. (2025). However, answering a question over long conversational histories poses a different challenge: evidence is often sparse and distributed across distant turns, and the question itself may omit the entities, events, or temporal cues needed to locate it Maharana et al. (2024); Wu et al. (2025). As a result, retrieving memories individually relevant to the question may still leave critical evidence undiscovered.

The key challenge is therefore not only which memories are relevant, but *what should be retrieved next given what has already been found* Huang & Huang (2026). Recovered evidence can supply search anchors absent from the original question Zhuang et al. (2024). Persistent-memory methods primarily maintain reusable memory, with less emphasis on a question-specific state of verified evidence Chhikara et al. (2025); Xu et al. (2025); Kang et al. (2025). IRCoT Trivedi et al. (2023) adapts queries using accumulated passages and reasoning, without an explicit verification step separating established evidence from other intermediate content. In methods that retrieve through graph traversal, access to further evidence depends partly on the preconstructed graph's connectivity Gutierrez et al. (2024); Pan et al. (2024); Li et al. (2024); Sun et al. (2024).

We introduce MERA (Missing-Evidence Retrieval Augmentation), which separates the question-specific evidence state from the global search space. The evidence graph organizes established facts within the planner's information state, helping identify what remains missing. As Figure 1 illustrates, newly established entities or temporal relations can guide retrieval over the global index, including evidence not yet represented in the graph. MERA pairs a global atomic-fact index with source-turn identifiers Chen et al. (2024) and an evidence graph for each question. A lightweight planner generates structured retrieval intents from the graph, question, and query history. A verifier gates which retrieved facts enter the graph to inform subsequent retrieval Wang et al. (2024); Yao et al. (2023). Retrieval continues until the evidence is judged sufficient, the search is exhausted, or

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.

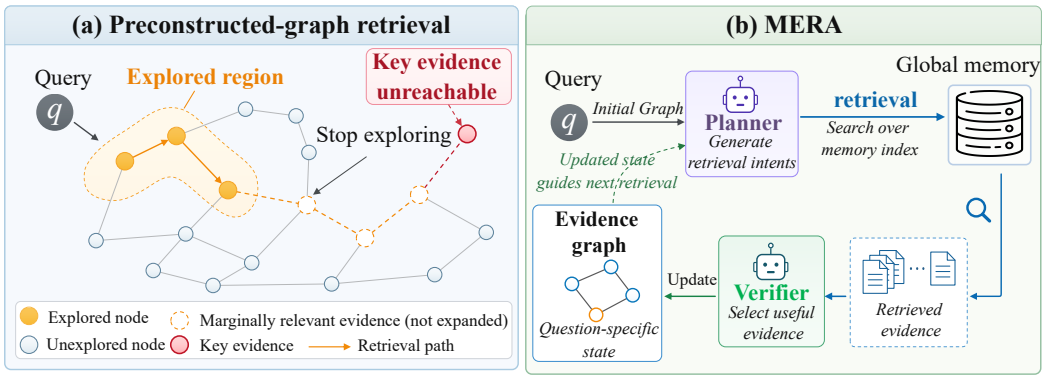

Figure 1: A question-specific evidence state should guide retrieval without defining the search boundary. (a) A preconstructed graph's explored region may exclude relevant evidence. (b) MERA searches a separate global memory index; verified evidence updates the state and guides subsequent retrieval. Orange and green diamonds mark required and verified evidence; panels are schematic.

the round budget is reached. The final graph and accepted evidence then provide the context for answer generation.

We train the planner to recover new evidence rather than reproduce a prescribed query. Differently worded queries may retrieve the same evidence, while a plausible query may only repeat earlier hits Zhu et al. (2025). We therefore evaluate actions by their retrieval outcomes Wu et al. (2022); Ma et al. (2023): for each intermediate evidence state, we execute multiple planner outputs against the memory index and reward newly recovered gold evidence, with no credit for repeated hits. Comparing queries from the same evidence state provides relative feedback on how effectively each action closes the remaining evidence gap. These rewards train the lightweight planner with Group Relative Policy Optimization (GRPO) Shao et al. (2024), while the memory index, evidence model, and answer model remain fixed.

MERA achieves strong answer accuracy across Qwen3-30B and GPT-4o-mini backbones. Using Qwen3-30B Team (2025) for evidence processing and answer generation, MERA's retrieval-trained 0.6B planner achieves 77.40% accuracy on LoCoMo and 71.29% on LongMemEval-S. It also exceeds a 30B planner without retrieval-grounded training by 4.10 and 3.96 percentage points, respectively. On LoCoMo, later retrieval rounds raise cumulative evidence recall from 55.5% to 80.5%.

Our contributions are threefold:

- We identify a query-time gap in long-term memory QA: persistent memory describes what can be retrieved, but answering a question additionally requires tracking what has been established and deciding what to retrieve next.
- We introduce MERA, which decouples a verifier-gated evidence graph from a globally searchable atomic-fact index, allowing verified evidence to guide search without restricting access to other relevant evidence.
- We train a lightweight planner using retrieval outcomes, rewarding previously missing evidence rather than supervising a unique query form. Under the evaluated settings, the trained 0.6B planner achieves higher answer accuracy than a 30B planner without retrieval-grounded training.

2 Related Work

2.1 Long-Term Memory for LLM Agents

Long-term memory systems mainly optimize persistent representation Zhang et al. (2025). Mem0 Chhikara et al. (2025) maintains salient facts through explicit add, update, delete, and no-op decisions; A-MEM Xu et al. (2025) organizes experiences as interconnected notes that evolve

with new memories. MemoryOS Kang et al. (2025) manages recent dialogue, topic-level pages, and personal knowledge in separate tiers, whereas LightMem Fang et al. (2025) compresses and consolidates interactions offline. These methods operate primarily on a store shared across future questions. MERA focuses on the retrieval decisions made while answering a particular question: it keeps the memory index fixed and maintains an explicit evidence state that lets the planner use verified facts and previous searches to identify missing information and decide what to retrieve next.

2.2 Learning to Manage and Use Memory

Memory-R1 Yan et al. (2026) applies reinforcement learning to persistent-memory operations and to selecting stored entries for answering. Lightweight LLM Agent Memory Zhang et al. (2026b) uses small models to coordinate retrieval, writing, and consolidation under a fixed retrieval budget, while MemSkill Zhang et al. (2026a) learns to select and evolve reusable memory-construction skills. MERA optimizes the query planner at intermediate evidence states, using the previously missing gold evidence turns recovered by each executed action as its primary reward. The learned action specifies what to search for next, conditioned on what earlier retrieval rounds have established; the persistent memory, evidence model, and answer model remain fixed during training.

2.3 Iterative Retrieval and Structured Evidence

IRCoT Trivedi et al. (2023) alternates retrieval with chain-of-thought generation, using the latest reasoning sentence as the next query and accumulated passages and thoughts as its search context. EfficientRAG Zhuang et al. (2024) trains lightweight token classifiers to retain useful passage tokens and assemble next-hop queries. FLARE Jiang et al. (2023) retrieves from low-confidence anticipated content, while Adaptive-RAG Jeong et al. (2024) routes questions among no-retrieval, single-step, and iterative strategies. These methods establish the value of adapting retrieval during answering Huang & Huang (2026). MERA addresses the next-query decision through an explicit, verifier-gated evidence state and trains the planner according to the missing evidence recovered by its actions, favoring queries that complement the facts already verified in earlier rounds.

GraphReader Li et al. (2024) constructs a document graph of key elements and atomic facts, then explores selected nodes, neighbors, and source chunks. Think-on-Graph Sun et al. (2024) searches an existing knowledge graph by iteratively selecting relations and entities with beam search, while HippoRAG Gutierrez et al. (2024) propagates query signals over a persistent corpus graph. These methods search graphs constructed independently of the current question. MERA instead retrieves from a fixed atomic-fact index and incrementally builds a transient graph from verified, source-linked evidence. The graph organizes the planner state without making evidence access depend on preconstructed graph connectivity, allowing subsequent queries to retrieve evidence about entities not yet represented in the graph or linked to its existing nodes.

3 Method

3.1 Problem Formulation

Let a long-term conversation history be a chronological sequence of turns $\mathcal{U} = \{u_1, \ldots, u_N\}$, where each turn includes its utterance, speaker, timestamp, and session metadata. Given a question q, the goal is to produce an answer supported by a sparse subset of turns. Because an entity or relation discovered in one turn may reveal what to retrieve next, the original question is often insufficient to recover the complete evidence chain in a single search.

We therefore formulate long-term memory QA as sequential evidence acquisition, where each retrieval decision depends on what has already been established and what the question still requires. At round t, a planner generates retrieval actions a_t from a question-specific evidence state s_t; a retriever executes them against persistent memory $\mathcal{M}$; and a verifier accepts useful evidence and updates the state:

$$a_t \sim \pi_\theta(\cdot \mid s_t), \quad C_t = \mathrm{Retrieve}(\mathcal{M}, a_t), \quad (E_t, \Delta G_t, b_t) = V_\phi(q, G_t, \mathcal{E}_t, C_t). \tag{1}$$

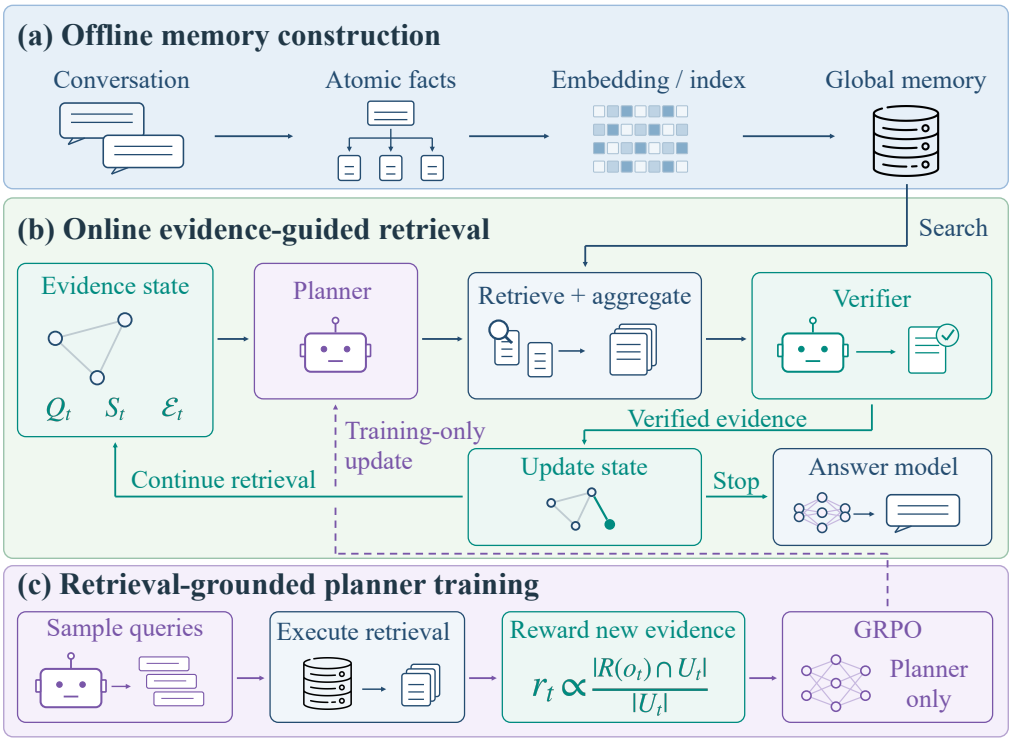

Figure 2: Overview of MERA. (a) Conversation turns are decomposed into atomic facts and indexed in global memory. (b) Planning, retrieval, aggregation, and verification update the evidence state for continued retrieval or answering. (c) Newly recovered gold evidence provides the reward for GRPO; the dashed arrow denotes the training-only planner update.

Here, C_t contains retrieved candidates, E_t is newly accepted evidence, ΔG_t is the graph update, and b_t indicates whether the evidence is sufficient. The objective is to acquire enough evidence for a correct answer within a bounded retrieval budget.

3.2 MERA Overview

MERA separates reusable memory from the evidence state maintained for each question through three stages (Figure 2). Offline, conversation turns are decomposed into self-contained atomic facts, embedded, and indexed with parent-turn identifiers. Online, a lightweight planner guides global retrieval using the evolving evidence state; parent-turn fact aggregation and verification supply evidence for further search or answering. As verified facts accumulate, they provide new entities and relations for subsequent queries, while global retrieval keeps evidence outside the current graph accessible. During training, multiple queries sampled from a shared state are executed against the same memory and rewarded for recovering previously missing gold evidence, giving the planner feedback on the actual retrieval outcomes. GRPO updates only the planner, while the memory index, evidence model, and answer model remain fixed.

3.3 Traceable Memory and Evidence State

For each conversation turn, MERA extracts self-contained atomic facts with retrieval metadata and embeds them for matching. A matched fact provides a focused similarity signal but may omit related information captured in other facts from the same turn, so before verification MERA aggregates all extracted facts belonging to each matched parent turn, together with its speaker, timestamp, and source identifier. We call this operation *parent-turn fact aggregation*. Results from multiple intents are merged and deduplicated by parent turn, giving the verifier a broader fact record, containing extracted facts rather than verbatim source text, while retaining provenance.

For question q, MERA represents the accumulated evidence in a transient graph G_t, within the complete evidence state s_t:

$$G_t = (V_t, L_t, X_t, D_t, \Psi_t), \qquad s_t = (q, G_t, Q_t, S_t, \mathcal{E}_t). \tag{2}$$

The graph stores verified entities and relations (V_t, L_t), active retrieval anchors X_t, exhausted or invalid directions D_t, and provenance links Ψ_t to source turns. Alongside the graph, the evidence-state box in Figure 2(b) shows previous queries Q_t, viewed candidates S_t, and accepted evidence $\mathcal{E}_t$. Together, these records let the planner compare what has been established and attempted against the question to identify what remains missing. The graph is constructed for the current question, while the persistent index determines which memories remain searchable. Appendices A.1 and A.2 detail atomic-fact construction, parent-turn fact aggregation, graph merging, and temporal normalization.

3.4 Evidence-State-Guided Active Retrieval

The evidence state is initialized from the question. At round t, the planner compares the information required by q with the current evidence state. If a gap remains, it emits a small set of structured retrieval intents,

$$a_t = \{I_t^\ell\}_{\ell=1}^{m_t}, \qquad I_t^\ell = (d_t^\ell, k_t^\ell, c_t^\ell), \tag{3}$$

where d_t^ℓ describes the desired fact, k_t^ℓ identifies salient entities or concepts, and c_t^ℓ supplies coarse topical labels. Conditioning on verified evidence and query history lets later intents use newly discovered anchors and avoid repeating earlier searches.

The retriever matches each intent against the global atomic-fact index, assembles the corresponding parent-level fact records, and removes previously viewed evidence. Candidate access does not require adjacency to a node in G_t: a semantic query can retrieve a fact about an entity absent from the current graph. The verifier admits only candidates that help resolve the question and updates the state as

$$G_{t+1} = \mathrm{Merge}(G_t, \Delta G_t), \quad \mathcal{E}_{t+1} = \mathcal{E}_t \cup E_t, \quad Q_{t+1} = Q_t \cup a_t. \tag{4}$$

The green node and edge in the *Update state* box illustrate newly verified information incorporated into the graph. The *Continue retrieval* branch returns the updated state to the planner. When evidence is sufficient, search is exhausted, or the round budget is reached, the *Stop* branch passes the final graph and accepted fact records, with their source identifiers, to the answer model. Appendix A.3 specifies the retrieval and termination rules; Appendix D.4 provides a logged trajectory, and Appendix C.2 reports planner runtimes and the inference environment.

3.5 Retrieval-Grounded Policy Optimization

Query generation does not have a unique correct string, so MERA evaluates a planner output by executing it against the actual memory. Planner-training questions come from dataset-specific sources: LoCoMo questions are synthesized with the benchmark's released data-generation utilities, while LongMemEval questions are generated from the designated training conversations. Training states are collected before planner calls throughout complete retrieval trajectories, spanning initial search, complementary retrieval, and late-stage completion; states with uniformly unsuccessful or uniformly complete outcomes are removed so optimization focuses on decisions where the query can change retrieval results. Appendix A.4.1 details the data construction process.

Let $\mathcal{E}^*$ be the gold parent turns, F_t the gold evidence found before round t, $U_t = \mathcal{E}^* \setminus F_t$ the remaining evidence, and $R(o_t)$ the parent turns retrieved by planner output o_t. We define

$$x_t = \frac{|R(o_t) \cap U_t|}{|U_t|}, \qquad p_t = \frac{|F_t|}{|\mathcal{E}^*|}, \qquad R_{\mathrm{ret}}(o_t) = x_t \left[1 + p_t(1 - x_t)\right]. \tag{5}$$

The fraction shown in Figure 2(c) is x_t, the proportion of missing gold evidence recovered by an action. The figure's $r_t \propto x_t$ is schematic: the implemented retrieval reward R_{ret} also includes the progress term above. Evidence already in F_t receives no additional credit, and queries with different wording receive the same retrieval reward when they recover the same missing gold turns. The progress term increases the reward for partial recovery of remaining evidence later in a trajectory, and parent-turn-level reward aligns training with gold annotations, verification, and answer generation. Auxiliary keyword-count and response-length penalties shape the planner output, as described in Appendix A.4.

We optimize the lightweight planner with Group Relative Policy Optimization (GRPO) Shao et al. (2024): for each shared evidence state, MERA executes a group of sampled outputs, normalizes the resulting rewards into relative advantages, and applies the clipped policy objective with KL regularization and an optional entropy term Schulman et al. (2017), so that with a shared question, graph, and memory across the group, relative rewards primarily reflect the sampled query actions. Appendix A.4 gives the complete trajectory fields, filtering rule, reward rationale, and a detailed derivation of the GRPO objective.

4 Experiments

We evaluate whether deciding what to retrieve next from accumulated evidence improves long-term memory QA. After reporting answer accuracy and token costs, we examine whether later queries recover missing evidence, how the evidence state supports retrieval and answering, and whether retrieval-grounded training improves the lightweight planner. The appendix examines parent-turn fact aggregation, query behavior, runtime, termination, and retrieval-hyperparameter sensitivity.

4.1 Experimental Setup

We use LoCoMo Maharana et al. (2024), which contains 1,986 questions, and LongMemEval-S Wu et al. (2025), which contains 500 questions. The compared systems are FullText, NaiveRAG Lewis et al. (2020), IRCoT Trivedi et al. (2023), GraphReader Li et al. (2024), LangMem LangChain (2025), A-MEM Xu et al. (2025), MemoryOS Kang et al. (2025), Mem0 Chhikara et al. (2025), LightMem Fang et al. (2025), and MERA-RL. We report results with Qwen3-30B-A3B-Instruct-2507 Team (2025) and GPT-4o-mini OpenAI (2024) as the cache-construction and inference backbone.

Answer accuracy is measured by an LLM judge. Generative-model costs include offline memory construction and online question answering but exclude embedding and vector retrieval. Appendix B provides dataset descriptions and splits, metric accounting, and implementation details.

4.2 Main Results

Table 1 jointly compares answer accuracy and offline memory-construction cost on LongMemEval-S and LoCoMo. We use three MERA configurations throughout the experiments. **MERA-Small** uses a Qwen3-0.6B query planner without retrieval-grounded training, while **MERA-RL** uses the same planner after retrieval-grounded GRPO Shao et al. (2024) training. In both configurations, the evidence model is the backbone specified in the corresponding table block. **MERA-Large** uses that table-specified backbone for both query planning and evidence processing, without retrieval-grounded planner training.

MERA-RL achieves the highest answer accuracy among the compared methods on both benchmarks with Qwen3-30B. With GPT-4o-mini, it also achieves the highest accuracy on LongMemEval-S and the strongest result among the memory-based systems on LoCoMo. The results show that MERA can recover and use relevant conversational evidence across both datasets and backbone settings.

This performance is consistent with the way MERA connects evidence acquisition to the current question state. Atomic facts provide focused retrieval targets, and parent-turn fact aggregation supplies related details for verification. The evidence graph then connects accepted facts across sessions and exposes entities, relations, and temporal anchors that can guide subsequent queries. As evidence accumulates, the planner can refine its search around information still needed to answer the question. Global retrieval keeps evidence accessible even when it concerns an entity absent from the

Table 1: Main results on LongMemEval-S and LoCoMo. ACC is answer accuracy. Offline generative-model costs are reported in thousands per question (k/Q) for LongMemEval-S and thousands per conversation (k/Conv.) for LoCoMo. Superscript * marks results obtained from our own runs.[1] "–" denotes not applicable or not reported; IRCoT makes no offline generative calls.

Backbone	Method	ACC (%)	Sum. In (k)	Sum. Out (k)	Upd. In (k)	Upd. Out (k)	Total (k)
			LongMemEval-S				
Qwen3-30B	FullText	54.80	–	–	–	–	105.07
	NaiveRAG	60.80	–	–	–	–	–
	IRCoT*	55.45	–	–	–	–	–
	GraphReader*	46.53	155.52	99.61	–	–	255.13
	LangMem	50.80	–	–	1,311.96	118.06	1,430.02
	A-MEM	65.20	219.21	66.98	1,260.54	318.20	1,864.93
	MemoryOS	49.60	2,101.54	510.88	305.12	27.43	2,944.97
	Mem0	39.51	424.20	15.34	411.50	111.35	962.39
	LightMem*	67.33	13.19	19.21	–	–	32.40
	MERA-RL*	**71.29**	328.74	91.97	–	–	420.70
GPT-4o-mini	FullText	56.80	–	–	–	–	105.07
	NaiveRAG	61.00	–	–	–	–	–
	LangMem	37.20	–	–	982.68	119.48	1,102.16
	A-MEM	62.60	214.66	42.82	1,157.52	190.81	1,605.81
	MemoryOS	44.80	2,302.35	304.18	350.02	35.19	2,991.74
	Mem0	53.61	424.13	17.76	560.17	150.56	1,152.62
	LightMem*	63.37	31.65	11.25	–	–	**42.91**
	MERA-RL*	**65.35**	313.83	83.23	–	–	397.07
			LoCoMo				
Qwen3-30B	FullText	74.87	–	–	–	–	–
	NaiveRAG	66.95	–	–	–	–	–
	IRCoT*	62.34	–	–	–	–	–
	GraphReader*	75.18	59.24	37.99	–	–	97.24
	LangMem	60.53	–	–	1,004.35	138.02	1,142.37
	A-MEM	56.10	158.29	60.85	924.19	483.51	1,626.84
	MemoryOS (LoCoMo)	61.04	122.21	53.12	104.43	81.75	361.51
	MemoryOS (regular)	51.30	228.85	51.60	242.27	143.63	666.35
	Mem0	43.31	827.09	18.64	763.88	189.80	1,799.41
	LightMem*	61.58	61.38	36.33	9.86	0.88	108.45
	MERA-RL*	**77.40**	148.32	19.73	–	–	168.05
GPT-4o-mini	FullText	**71.83**	–	–	–	–	–
	NaiveRAG	63.64	–	–	–	–	–
	LangMem	57.20	–	–	898.27	111.95	1,010.22
	A-MEM	64.16	182.74	49.29	729.89	187.52	1,149.44
	MemoryOS (LoCoMo)	58.25	110.98	33.40	78.08	64.54	287.00
	MemoryOS (regular)	54.87	226.86	46.61	177.66	75.34	526.47
	Mem0	61.69	851.32	20.53	632.12	189.42	1,693.39
	LightMem*	63.29	**62.82**	**17.95**	4.14	0.28	**85.19**
	MERA-RL*	**67.02**	270.07	36.79	–	–	306.86

current graph. These properties are useful for long histories, where the initial question may omit the intermediate clues needed to locate supporting evidence. Section 4.5 and the ablations below examine this process through evidence recovery and component comparisons.

MERA combines this accuracy with lower offline token consumption than most memory systems in Table 1. Its construction cost is concentrated in extracting atomic facts from conversation turns; the resulting index is reusable across questions. The pipeline processes each turn without repeatedly presenting an expanding memory to a generative model for consolidation or rewriting. Question-specific evidence organization takes place during retrieval, so the offline stage need not anticipate every relation that a future question may require. This division of work helps limit memory-maintenance overhead while retaining fine-grained evidence for later search.

4.3 Does Retrieval Feedback Improve a Lightweight Planner?

We compare MERA-Small (0.6B) and MERA-Large (30B), both without retrieval-feedback training, with MERA-RL, the trained 0.6B planner. All three use Qwen3-30B for evidence processing and answering; we measure answer accuracy.

[1] Results marked with * are obtained from our own runs under the evaluation setup described above, including our IRCoT, GraphReader, LightMem, and MERA-RL evaluations.

4.3.1 PLANNER SCALE AND POLICY-TRAINING ABLATION

Table 2: Planner-scale and policy-training ablation: answer accuracy (%).

Planner	LoCoMo	LongMemEval-S
MERA-Small	68.40	59.80
MERA-Large	73.30	67.33
MERA-RL	**77.40**	**71.29**

Table 2 shows that both training and scaling improve answer accuracy. Relative to MERA-Small, MERA-Large gains 4.90 points on LoCoMo and 7.53 points on LongMemEval-S, while MERA-RL gains 9.00 and 11.49 points, respectively. The trained 0.6B planner therefore exceeds MERA-Large by 4.10 and 3.96 points. The training objective rewards recovery of evidence missing from the current state, encouraging queries that complement earlier findings. These results show that, under the shared evidence and answer models, training the lightweight planner yields larger accuracy gains than increasing planner size alone.

Additional analyses. Appendix D examines query behavior and parent-turn fact aggregation. Appendix E analyzes question types and failures, while Appendix F reports retrieval-hyperparameter sensitivity.

4.4 ONLINE INFERENCE COST

With Qwen3-30B, MERA's online token consumption is on the order of tens of thousands of tokens per question and remains below the graph-navigation baseline on both datasets. This cost accompanies the leading answer accuracy reported in Table 1. The online computation is directed toward successive planning and verification over retrieved evidence. Atomic retrieval selects candidate records, verification filters their relevance, and the evolving graph organizes accepted facts for subsequent queries and final answering. This process focuses model context on evidence related to the question, while allowing additional retrieval when the current evidence is incomplete. Appendix C.1 provides the input/output token breakdown for each dataset.

4.5 HOW DOES MULTI-ROUND RETRIEVAL IMPROVE EVIDENCE COVERAGE?

MERA is designed to use the current evidence state to guide follow-up retrieval for information missed in earlier rounds. We measure how much additional gold evidence is recovered after the first round by tracking cumulative turn-level recall throughout retrieval.

Later rounds recover complementary evidence. On LoCoMo, recall after the first round increases by 14.1, 25.0, and 15.6 percentage points for MERA-Small, MERA-RL, and MERA-Large, respectively. MERA-RL rises from 0.555 to 0.805, showing that later actions recover substantial complementary evidence. On LongMemEval-S, MERA-RL rises from 0.849 to 0.989, a further 14.0-point gain. Among questions with a hit, the first gold evidence is recovered at round 1.13 on average, so additional rounds primarily complete the evidence set. These results support complementary retrieval; Section 4.6 examines the contribution of the graph-based reasoning process.

Most evidence gain occurs early: at least 92% of the LoCoMo curves' post-first-round improvement is realized by round 5. Appendix D.1 discusses diminishing returns and retriever compatibility; Appendix C.3 analyzes stopping behavior.

4.6 HOW DOES THE EVIDENCE STATE SUPPORT RETRIEVAL AND ANSWERING?

We compare the graph-assisted evidence workflow with a flat-evidence variant that removes the evidence graph and its associated state-maintenance mechanisms. Accepted evidence accumulates in a flat list for planning and final answering; atomic retrieval and parent-turn fact aggregation remain enabled. All no-graph configurations use Qwen3-30B for evidence processing and answering. The no-graph runs use dense atomic-fact retrieval with MMR ($\lambda = 0.95$), $k = 8$, and a budget of ten rounds. The planners are the base 0.6B model, its retrieval-trained counterpart, and the 30B model. Figure 3(c)–(d) compares these workflows, rather than isolating graph representation alone. Appendix D.2 reports the experimental configuration and results.

Graph-assisted configurations attain higher accuracy in all six comparisons. The graph–flat gaps for MERA-Small, MERA-RL, and MERA-Large are 14.02, 14.41, and 9.65 points on LoCoMo,

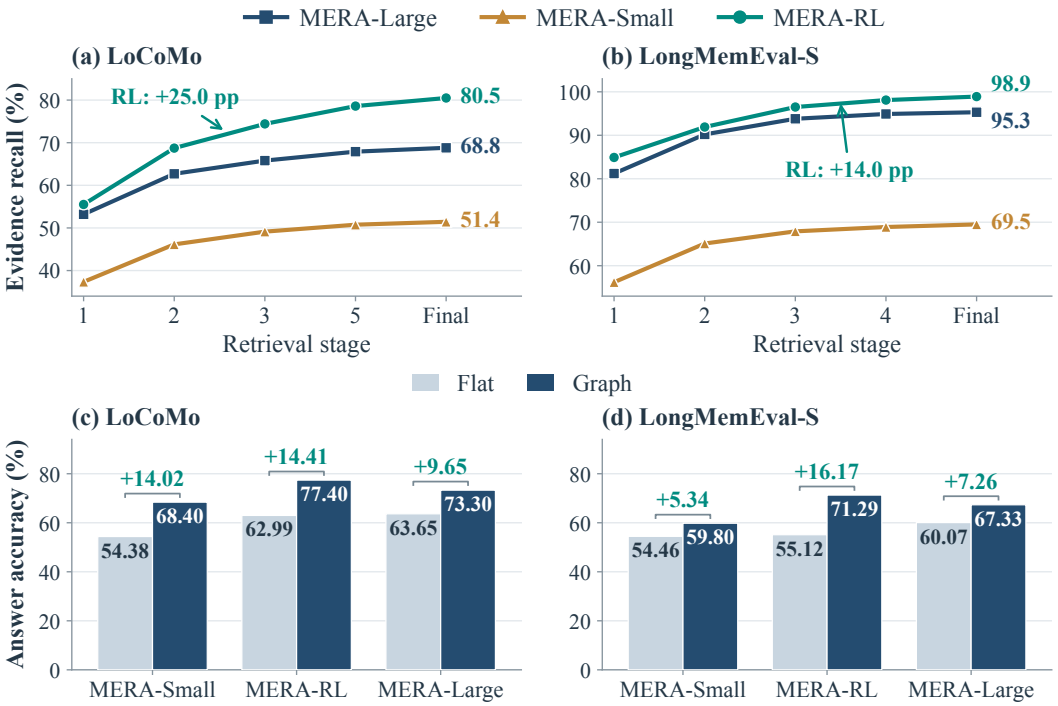

Figure 3: Evidence recall and graph ablations. (a)–(b) Cumulative recall with dataset-specific y-axis ranges; arrows show RL gains from round 1 to termination. (c)–(d) Answer accuracy; brackets show Graph minus Flat. All gains are in percentage points.

respectively; the corresponding gaps on LongMemEval-S are 5.34, 16.17, and 7.26 points. All graph scores use the corresponding planner configurations and answer accuracies reported in Table 2. On LoCoMo, the Small and RL gaps are similar, whereas on LongMemEval-S the RL gap is larger than those of the other two planners. The magnitude of the difference therefore varies across both datasets and planner configurations. Positive gaps occur for both planners without retrieval-feedback training as well as the RL planner, so the observed pattern is not confined to the trained policy.

The planner ranking also depends on the evidence workflow. With flat evidence, MERA-Large exceeds MERA-RL by 0.66 points on LoCoMo and 4.95 points on LongMemEval-S. With the graph-assisted workflow, MERA-RL instead leads by 4.10 and 3.96 points. Thus, the trained lightweight planner's advantage over the large planner in Table 2 is observed with the graph-assisted workflow and does not persist in this flat-evidence variant. This comparison characterizes the performance of the evaluated configurations without assigning the differences to a specific graph mechanism.

5 Conclusion

Long-term memory QA requires deciding what to retrieve next given what has already been found. MERA separates reusable global memory from a question-specific, verifier-gated evidence state and trains a lightweight planner on newly recovered evidence. MERA achieves strong answer accuracy on LoCoMo and LongMemEval-S with both Qwen3-30B and GPT-4o-mini backbones. With Qwen3-30B for evidence processing and answering, its trained 0.6B planner also surpasses a 30B planner without retrieval-grounded training in answer accuracy, while later retrieval rounds recover evidence missed initially. These results support evidence-guided retrieval for long-term memory QA and suggest that, under the evaluated settings, training a lightweight planner can be more effective than increasing planner size without retrieval-grounded training.

AI USE STATEMENT

We used generative AI tools to polish drafts written by the authors, including improving grammar, phrasing, clarity, and the flow of the text, and checking consistency across sections. We also used AI assistance to refine the visual presentation of figures and to adjust manuscript formatting and layout. For literature retrieval and discovery, we used AI tools to help find additional studies addressing similar problems or using related methods; these works are discussed in the Related Work section. A code agent assisted with writing and debugging experiment code.

The research ideas, method design, and experimental design are entirely our own; no generative AI tool was used to generate research ideas. We have reviewed all AI-assisted content: the manuscript text was checked paragraph by paragraph by the authors for accuracy and clarity, and AI-assisted code was verified by the authors and tested in our actual experiments. We take full responsibility for the final content of this work, including the text, experimental claims, and any artifacts produced with the aid of generative AI tools.

ETHICS STATEMENT

This work uses LoCoMo and LongMemEval-S, two publicly released long-term conversational question-answering benchmarks whose dialogues are synthetic or de-identified and do not contain identifiable private information about real users. This study does not involve new human-subject experiments and therefore did not require institutional review board approval. We note that a retrieval-augmented long-term memory system deployed in a real product would need to carefully address user conversation privacy and controllable management or deletion of stored memories; this paper, however, focuses on the retrieval and planning algorithm itself and does not address a specific product deployment. We are not aware of any funding conflicts, discriminatory bias, or other issues in this work that would raise concerns under the Code of Ethics.

REPRODUCIBILITY STATEMENT

We will publicly release the complete code repository and training data once this paper has passed peer review, so that our experimental results can be independently reproduced. The complete implementation details, hyperparameter configurations, and the retrieval-verification procedure are given in Section 3 and the appendix; complete dataset descriptions, metric accounting, and experimental settings are given in Appendix B; and the construction of retrieval-grounded policy-training data is detailed in Appendix A.4.1. All backbone models used in this work are publicly available: Qwen3-30B and Qwen3-0.6B are open-source models, GPT-4o-mini is accessed through a public API, and the embedding model used for retrieval is likewise publicly released, facilitating reproduction of our results under the same conditions.

REFERENCES

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. Dense X retrieval: What retrieval granularity should we use? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 15159–15177. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.845. URL https://doi.org/10.18653/v1/2024.emnlp-main.845.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In Inês Lynce, Nello Murano, Mauro Vallati, Serena Villata, Federico Chesani, Michela Milano, Andrea Omicini, and Mehdi Dastani (eds.), *ECAI 2025 - 28th European Conference on Artificial Intelligence, 25-30 October 2025, Bologna, Italy - Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025)*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pp. 2993–3000. IOS Press, 2025. doi: 10.3233/FAIA251160. URL https://doi.org/10.3233/FAIA251160.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. Lightmem: Lightweight and efficient memory-augmented generation. *CoRR*, abs/2510.18866, 2025. doi: 10.48550/ARXIV.2510.18866. URL https://doi.org/10.48550/arXiv.2510.18866.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html.

Yizheng Huang and Jimmy Xiangji Huang. A survey on retrieval-augmented text generation for large language models. *ACM Comput. Surv.*, 58(12), May 2026. ISSN 0360-0300. doi: 10.1145/3805774. URL https://doi.org/10.1145/3805774.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7036–7050, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.389. URL https://aclanthology.org/2024.naacl-long.389/.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.495. URL https://aclanthology.org/2023.emnlp-main.495/.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 25961–25970. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.1318. URL https://doi.org/10.18653/v1/2025.emnlp-main.1318.

LangChain. Langmem sdk for agent long-term memory, 2025. URL https://blog.langchain.com/langmem-sdk-launch/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, Wenbo Su, and Bo Zheng. Graphreader: Building graph-based agent to enhance long-context abilities of large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pp. 12758–12786. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.746. URL https://doi.org/10.18653/v1/2024.findings-emnlp.746.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 5303–5315. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.322. URL https://doi.org/10.18653/v1/2023.emnlp-main.322.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 13851–13870. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.747. URL https://doi.org/10.18653/v1/2024.acl-long.747.

OpenAI. GPT-4o mini: Advancing cost-efficient intelligence, 2024. URL https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Trans. Knowl. Data Eng.*, 36(7): 3580–3599, 2024. doi: 10.1109/TKDE.2024.3352100. URL https://doi.org/10.1109/TKDE.2024.3352100.

Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In Sean Follmer, Jeff Han, Jürgen Steimle, and Nathalie Henry Riche (eds.), *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST 2023, San Francisco, CA, USA, 29 October 2023- 1 November 2023*, pp. 2:1–2:22. ACM, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

James Pustejovsky, José M. Castaño, Robert Ingria, Roser Saurí, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. Timeml: Robust specification of event and temporal expressions in text. In Mark T. Maybury (ed.), *New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium, Stanford University, Stanford, CA, USA*, pp. 28–34. AAAI Press, 2003.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Jannik Strötgen and Michael Gertz. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In Katrin Erk and Carlo Strapparava (eds.), *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010*, pp. 321–324. The Association for Computer Linguistics, 2010. URL https://aclanthology.org/S10-1071/.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=nnVO1PvbTv.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In Anna

Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 10014–10037. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.557. URL https://doi.org/10.18653/v1/2023.acl-long.557.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers Comput. Sci.*, 18(6): 186345, 2024. doi: 10.1007/S11704-024-40231-1. URL https://doi.org/10.1007/s11704-024-40231-1.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. CONQRR: conversational query rewriting for retrieval with reinforcement learning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 10000–10014. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.679. URL https://doi.org/10.18653/v1/2022.emnlp-main.679.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for LLM agents. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.583. URL https://aclanthology.org/2026.acl-long.583/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. Memskill: Learning and evolving memory skills for self-evolving agents. *CoRR*, abs/2602.02474, 2026a. doi: 10.48550/ARXIV.2602.02474. URL https://doi.org/10.48550/arXiv.2602.02474.

Jiaquan Zhang, Chaoning Zhang, Shuxu Chen, Zhenzhen Huang, Pengcheng Zheng, Zhicheng Wang, Ping Guo, Fan Mo, Sung-Ho Bae, Jie Zou, Jiwei Wei, and Yang Yang. Lightweight LLM agent memory with small language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 12914–12929. Association for Computational Linguistics, 2026b. doi: 10.18653/V1/2026.ACL-LONG.588. URL https://doi.org/10.18653/v1/2026.acl-long.588.

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.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 19724–19731. AAAI Press, 2024. doi: 10.1609/AAAI.V38I17.29946. URL https://doi.org/10.1609/aaai.v38i17.29946.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. Large language models for information retrieval: A survey. *ACM Trans. Inf. Syst.*, 44(1), November 2025. ISSN 1046-8188. doi: 10.1145/3748304. URL https://doi.org/10.1145/3748304.

Ziyuan Zhuang, Zhiyang Zhang, Sitao Cheng, Fangkai Yang, Jia Liu, Shujian Huang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. Efficientrag: Efficient retriever for multi-hop question answering. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 3392–3411. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.199. URL https://doi.org/10.18653/v1/2024.emnlp-main.199.

APPENDIX

A METHOD DETAILS

A.1 ATOMIC-FACT MEMORY AND PARENT-TURN FACT AGGREGATION

For each conversation turn, an extraction model produces self-contained atomic facts together with keywords and topic tags. Speaker identity, timestamp, and turn identifier are retained as metadata. Pronouns are replaced with explicit entities whenever possible, and subject–action–object directionality is preserved to reduce ambiguity outside the original context. Each fact is embedded together with its keywords and topics, while its parent identifier and the full set of facts extracted from that turn are stored alongside the vector. This memory index is constructed offline and reused across questions associated with the same conversation history.

Matching individual facts provides a focused retrieval signal, while related facts extracted from the same turn can supply additional information for verification. MERA therefore uses fine-to-coarse parent–child retrieval: atomic facts are the matching units, while parent-level fact records are the verification and reasoning units. Each record contains all extracted facts from one turn together with speaker and timestamp metadata. The source-turn identifier preserves provenance; the text presented to the verifier is the extracted fact set, not the verbatim conversation turn.

For each retrieval intent, MERA retrieves the top-k atomic candidates, merges results across intents, and deduplicates them by parent turn. When several facts from the same turn are retrieved, their parent record appears only once and is ranked by its highest atomic matching score. The verifier receives the aggregated fact record with speaker and temporal metadata. Viewed fact and parent-turn identifiers are added to an exclusion set so that the same evidence does not repeatedly enter later verification rounds.

A.2 EVIDENCE-GRAPH CONSTRUCTION AND NORMALIZATION

The graph implements the structured record within the question-specific evidence state. In $G_t = (V_t, L_t, X_t, D_t, \Psi_t)$, V_t contains typed nodes for people, organizations, locations, events, dates, quantities, and states. L_t is a set of directed labeled relations; X_t records active entities that can anchor further retrieval; D_t stores search directions invalidated or exhausted by the current evidence; and Ψ_t maps graph elements to their supporting parent turns. The complete evidence state $s_t = (q, G_t, Q_t, S_t, \mathcal{E}_t)$ also contains previous queries Q_t, viewed candidates S_t, and accumulated accepted evidence $\mathcal{E}_t$. These records guide the next query; they do not define the contents or connectivity of the searchable memory.

Before retrieval, the evidence model extracts a small set of salient people, events, locations, or concepts from the question as seed nodes. If structured parsing fails, MERA uses the complete question as a fallback node. In later rounds, the verifier converts only directly useful candidates into entities, attributes, and labeled relations, retaining provenance for every new node and edge. Graph merging reuses matching entities, removes duplicate relations, and assigns versioned identifiers to conflicting attributes so that states from different times can coexist. A direction is marked invalid only when new evidence explicitly contradicts it or establishes that it no longer holds.

A conversation timestamp indicates when a dialogue turn occurred, not necessarily when the described event occurred. The verifier therefore distinguishes turn time from event time: exact relative expressions are resolved against the turn timestamp, vague expressions retain anchored intervals, and the turn timestamp is not used as event time when no temporal expression is present Pustejovsky et al. (2003); Strötgen & Gertz (2010). The same explicit calculation process handles ages, amounts, counts, and durations. Normalized values are stored with their original statements and provenance.

A.3 RETRIEVAL EXECUTION, VERIFICATION, AND TERMINATION

Each structured intent $I_t^\ell = (d_t^\ell, k_t^\ell, c_t^\ell)$ specifies the desired fact, key entities or concepts, and topical labels. The planner compares the requirements of the question with the evidence state to identify missing entities, relations, dates, or quantities. Previous queries are explicitly included in the state,

Table 3: Construction and filtering of planner-training data. A state is the input saved before one planner call in a retrieval trajectory. Retention is measured against the states entering rollout filtering.

Dataset	Histories	Generated QAs	Collected states	Rollout input	Retained
LoCoMo	170	24,787	74,742	74,742	2,996 (4.0%)
LongMemEval	277	11,017	17,752	14,475	2,312 (16.0%)

encouraging alternative expressions and search directions when earlier actions have not resolved a gap. The retriever concatenates the intent fields into query text and searches the global atomic-fact index, excluding previously viewed evidence. It does not traverse the evidence graph to enumerate candidates, so retrieved facts can introduce entities not yet recorded in that graph.

After the retriever returns parent-level fact records, the evidence model verifies each candidate using the question, current graph, and previously accepted evidence. Records judged useful for resolving the question enter the accepted set. Subsequent planning therefore depends on verified evidence rather than unfiltered retrieval text.

The loop terminates when the planner or verifier judges the evidence sufficient after a minimum exploration requirement, when several consecutive rounds yield no valid intent, candidate, or graph update, or when the process reaches $T_{\max}$. The answer model then generates an answer from the question, final graph, and accumulated accepted fact records. If retrieval stops because of exhaustion or the round limit, the model answers only from the currently verified evidence.

A.4 Planner Training Details

A.4.1 Training Data Construction

For LoCoMo, we use the data-generation utilities released in the benchmark repository. The generator samples detailed personas and dated personal events, organizes those events into multi-session schedules, and produces long conversations conditioned on the resulting profiles and timelines. A question generator then reads each complete conversation and creates questions together with answers and supporting dialogue turns. This procedure provides many distinct conversational histories while preserving the persona, event, and temporal structure needed for long-term memory retrieval.

For LongMemEval, we use conversations from our training partition, defined in Appendix B, rather than synthesizing new histories. We generate additional questions from each training conversation, require every answer to be grounded in the conversation, and align the supporting evidence with retrievable parent turns. The two procedures therefore differ only in how the underlying questions and conversations are obtained: LoCoMo supplies a repository-native pipeline for generating both, whereas LongMemEval supplies the training conversations and MERA augments them with additional questions.

Table 3 summarizes the resulting data. The LoCoMo pipeline generates both the conversation histories and their questions. For LongMemEval, the question count combines the two augmentation rounds used by the final training set. Its collected-state count includes all trajectory states from these rounds; evidence validation and turn alignment are applied before rollout filtering, which reduces the number of states entering that stage.

A.4.2 Multi-Round State Collection and Filtering

After question construction, both datasets use the same trajectory collection and filtering procedure. Training states are collected from complete MERA trajectories. Before every planner call, the system stores the question, evidence graph, query history, complete gold evidence set $\mathcal{E}^*$, gold evidence already found F_t, and the corresponding memory. The data therefore cover initial retrieval, complementary retrieval from partial evidence, and late-stage evidence completion.

For each state, the base planner generates multiple rollouts whose queries are executed against the actual memory. A state is too difficult for the current policy when every rollout retrieves no new evidence, and too easy when every rollout retrieves all remaining evidence. MERA removes both

cases and retains states with mixed or partial outcomes, concentrating training on decisions for which policy choice can change the retrieval result.

A.4.3 Incremental Turn-Level Retrieval Reward

The retrieval reward is defined in Section 3.5. It evaluates whether a query finds evidence still missing from the current state. Evidence already found receives no additional reward. At the beginning of retrieval, R_{ret} reduces to recall over the remaining evidence. As search progresses, the progress term increases the reward for partial recovery of the remaining evidence. Reward is computed at the parent-turn level, matching the evidence unit used by verification and inference.

The reward has five useful properties. First, it provides action-level feedback immediately after query execution, shortening the credit-assignment path compared with final-answer reward. Answer correctness is jointly affected by evidence verification, temporal calculation, and answer generation, whereas newly recovered evidence more directly measures the quality of the current query. Second, the reward evaluates retrieval outcomes rather than similarity to a prescribed query string. Multiple query formulations receive equivalent feedback when they recover the same target evidence, which is important because effective queries are not unique.

Third, only uncovered evidence contributes to reward. Repeatedly retrieving an existing turn receives no additional credit, shifting the objective from maximizing raw hits to reducing the current evidence gap and encouraging complementary actions across rounds. Fourth, for a fixed partial-recovery fraction $0 < x_t < 1$, the progress term $p_t(1 - x_t)$ gives additional weight to actions taken after more gold evidence has already been found. This term encourages continued evidence acquisition from incomplete states later in a trajectory. Fifth, reward is computed at the parent-turn level, the same unit used for gold annotations, evidence verification, and answer generation. This alignment reduces a granularity mismatch between training and inference.

The reward complements informative-state filtering. Filtering first removes states in which policy choice cannot change the outcome, and incremental reward then distinguishes how much the remaining actions reduce the evidence gap. Training computation is therefore concentrated on decision points with meaningful policy variation rather than diluted by uniformly successful or uniformly unsuccessful samples.

We additionally regularize planner outputs with a keyword-count penalty that discourages overly broad retrieval intents and a response-length penalty that discourages excessively short or verbose actions. These auxiliary terms shape the form of the retrieval action, while the executed retrieval outcome remains the primary learning signal.

A.4.4 Policy Optimization

We train the planner with Group Relative Policy Optimization (GRPO). Unlike actor–critic methods, GRPO does not fit a separate value model. For a planner state s, the old policy π_{old} samples a group of G responses $o_1, \ldots, o_G$. MERA parses and executes every response against the same memory and assigns it a scalar reward r_i. The group mean and standard deviation define the relative advantage

$$\bar{r} = \frac{1}{G}\sum_{i=1}^{G} r_i, \qquad \sigma_r = \sqrt{\frac{1}{G-1}\sum_{i=1}^{G}(r_i - \bar{r})^2}, \qquad A_i = \frac{r_i - \bar{r}}{\sigma_r + \epsilon}. \tag{6}$$

Thus, an output receives a positive advantage only when it performs better than other outputs generated for the same evidence state. This within-group comparison controls for variation in question and state difficulty without requiring a learned critic. The same sequence-level advantage A_i is applied to every generated token in o_i. For token $o_{i,j}$, let

$$\rho_{i,j}(\theta) = \frac{\pi_\theta(o_{i,j} \mid s, o_{i,<j})}{\pi_{\text{old}}(o_{i,j} \mid s, o_{i,<j})}. \tag{7}$$

GRPO maximizes the token-aggregated clipped surrogate

$$J_{\text{clip}}(\theta) = \frac{1}{\sum_i |o_i|} \sum_{i=1}^{G} \sum_{j=1}^{|o_i|} \min(\rho_{i,j}(\theta) A_i, \text{clip}(\rho_{i,j}(\theta), 1-\varepsilon, 1+\varepsilon)\, A_i)\,. \tag{8}$$

Clipping limits destructive policy updates when the new policy moves too far from the rollout policy. We additionally regularize the planner toward a fixed reference policy and optionally encourage exploration through token entropy:

$$J(\theta) = J_{\text{clip}}(\theta) - \beta D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}}) + \eta\, \mathcal{H}(\pi_\theta), \tag{9}$$

where β and η control the KL and entropy terms, respectively, and η may be zero. Because all members of a group share the question, graph, query history, and memory, reward differences primarily measure the quality of their query actions. If every response in a group obtains the same reward, its normalized advantages vanish and the state contributes no comparative learning signal. The informative-state filtering described above removes states that the base planner solves uniformly or fails uniformly, increasing the frequency of groups with meaningful reward variation. GRPO then shifts probability mass toward executable query actions that recover more of the missing evidence while constraining policy drift.

A.4.5 Scope and Future Directions of RL Training

Our RL design is deliberately simple: a lightweight planner is trained with GRPO using incremental evidence recovery as the main reward, while the memory index, evidence model, and answer model remain fixed. The trained-versus-untrained comparison supports the effectiveness of this training setup as a whole.

Future work could compare reward variants under matched training data and compute budgets, including evidence recall alone, incremental recall without progress weighting, and combinations of retrieval feedback and final-answer correctness. Ablating the auxiliary penalties and state filtering separately would help identify which components contribute to the observed gains. Beyond these comparisons, cost-aware rewards could balance evidence acquisition against retrieval rounds and token use, while trajectory-level optimization could jointly train query selection and stopping decisions. These extensions would examine whether improvements in immediate evidence recovery also translate into more efficient and accurate multi-round reasoning.

B Expanded Experimental Setup

B.1 Datasets

LoCoMo. LoCoMo contains 10 long conversations and 1,986 questions. Each conversation records two speakers interacting across multiple time points and includes personal events, relations, preferences, and evolving states. The five question categories are multi-hop, temporal, open-domain, single-hop, and adversarial. LoCoMo therefore evaluates factual retention, cross-session evidence composition, temporal reasoning, and responses to questions whose answers are unsupported by the conversation.

LongMemEval-S. LongMemEval-S contains 500 questions for evaluating whether conversational assistants can retain, retrieve, and use information from long interaction histories. Each question is associated with multiple historical sessions, of which only a small subset supports the answer. Its categories include single-session user and assistant information, user preferences, multi-session integration, temporal reasoning, and knowledge updates. The benchmark emphasizes sparse evidence localization under substantial historical distraction.

The datasets are complementary: LoCoMo emphasizes information evolution in continuous conversations, whereas LongMemEval-S emphasizes sparse evidence in high-noise histories. We evaluate answer accuracy on all 1,986 LoCoMo questions. All LongMemEval-S test evaluations reported in this paper use the same held-out test split of 101 questions, including the main comparisons, ablations, and supplementary analyses. According to the data split, instances outside the test set are used for data generation, as described in Appendix A.4.1.

B.2 Compared Methods

The main comparison includes FullText, NaiveRAG, IRCoT, GraphReader, LangMem, A-MEM, MemoryOS, Mem0, and LightMem. FullText directly supplies the complete history to the answer model, while NaiveRAG performs one-shot similarity retrieval. IRCoT and GraphReader provide iterative retrieval comparisons. The remaining systems represent structured, dynamically updated, or compressed long-term memory. These comparisons examine both persistent memory design and how retrieval proceeds while answering a question. We report results with Qwen3-30B-A3B-Instruct-2507 and GPT-4o-mini as the backbone.

Unless stated otherwise, we use Qwen3-30B-A3B-Instruct-2507 for cache construction, seed-graph generation, evidence verification and graph update, and final answer generation. In the GPT-4o-mini experiments, we rebuild the cache with GPT-4o-mini and use it for the same inference stages while holding the planner configuration fixed.

B.3 Metrics

Answer accuracy. We use LLM-as-a-Judge Accuracy as the primary task metric. The judge reads the question, reference answer, and model answer and determines whether the response is correct. Accuracy is the proportion of correctly answered questions in the evaluation set.

Generative-model cost. We report generative-model token usage separately for offline memory construction and online question answering. Summary In and Summary Out measure the input and output tokens used to transform raw conversations into retrievable memories. Update In and Update Out measure persistent memory-update tokens. Their sum is reported as Total. MERA performs atomic-fact extraction but no separate persistent update, so its Update entries are denoted by "–" rather than zero.

LongMemEval-S provides a separate history for each question, so offline costs are averaged per question (k/Q). In LoCoMo, one memory is shared by all questions associated with a conversation, so offline costs are averaged over the 10 conversations (k/Conv.). Online In, Out, and Total are averaged per question for both datasets. Embedding and vector retrieval do not invoke a generative model and are excluded from token counts.

B.4 Implementation Details

MERA uses a query planner to generate retrieval intents from the current evidence state and BAAI/bge-large-en-v1.5 to embed atomic facts and retrieval intents. The main results on both datasets use MERA-RL, with the cache-construction and inference backbone set as described above.

We adapt IRCoT and GraphReader to the same conversational question sets using Qwen3-30B-A3B-Instruct-2507 for generation and judging. IRCoT retrieves six turns per BM25 query, accumulates at most 15 passages of up to 350 words each, and generates at most ten reasoning sentences at temperature zero. GraphReader uses session-dated histories, sentence-aligned chunks of 1,000 characters for LoCoMo and 4,000 for LongMemEval-S, and local BGE-large-en-v1.5 embeddings. Its fact-exploration and chunk-reading agents allow at most 12 and 16 calls, respectively, at temperature 0.4. The larger LongMemEval-S chunk size is a resource adjustment in this adaptation.

For these two baselines, costs use the input and output usage returned by each generative API call, including generated text discarded by subsequent processing and logical prompt tokens served from a prefix cache. Judge calls are excluded. Exact replay of saved responses during recovery is counted once; interrupted requests that returned no usage are not measurable and are excluded. Neither baseline performs a separate persistent memory update.

C Online Cost, Runtime, and Termination Diagnostics

C.1 Online Inference Cost

Tables 4 and 5 report the input and output token costs underlying Section 4.4, using Qwen3-30B throughout. Both datasets use thousands of tokens per question (k/Q), excluding offline construction,

Table 4: Online inference cost on LoCoMo, in thousands of tokens per question (k/Q).

Method	In (k)	Out (k)	Total (k)
IRCoT	4.29	0.13	4.43
GraphReader	24.50	0.54	25.04
MERA	13.47	1.80	15.27

Table 5: Online inference cost on LongMemEval-S, in thousands of tokens per question (k/Q).

Method	In (k)	Out (k)	Total (k)
IRCoT	19.02	0.24	19.26
GraphReader	32.82	0.52	33.34
MERA	14.97	2.43	17.40

embeddings, and judging. MERA's online cost includes seed-graph construction, query planning, evidence verification and graph update, and final answer generation.

C.2 Runtime Statistics

We measure online runtime on all 101 LongMemEval-S test questions using a shared prebuilt memory cache on an NVIDIA GeForce RTX 4090 server. Qwen3-30B-A3B-Instruct-2507 is served by SGLang with tensor parallelism across four GPUs, while Qwen3-0.6B is served by a separate SGLang instance on one GPU. The evaluation uses sample-level parallelism of 32. Runtime per question includes query planning, retrieval, evidence verification and graph update, and final answer generation.

Policy training increases runtime by only 5.1 seconds per question because MERA-RL retains the same 0.6B planner architecture and serving setup as MERA-Small. MERA-Large is substantially slower because every planning step invokes the 30B service and its measured trajectories contain more LLM calls. These results show that retrieval-grounded training improves the behavior of the lightweight planner without incurring the inference cost of replacing it with the large backbone.

C.3 Termination Behavior

Figure 4 shows that most questions terminate well before the maximum retrieval budget. On LoCoMo, MERA-Small averages 3.15 rounds yet terminates without complete gold evidence on 52.5% of questions. MERA-RL occupies a substantially better operating point, with 2.20 rounds and 27.2% incomplete recall, while MERA-Large reaches 2.27 rounds and 36.2%. For MERA-Small, verifier-declared convergence accounts for 78.8% of stops, while retrieval exhaustion accounts for 20.5%. These stopping frequencies accompany a final evidence recall of 0.514. They motivate examining whether continued retrieval could recover useful evidence, but neither incomplete gold recall nor the aggregate stopping frequencies establish that a particular sufficiency decision was incorrect: some questions can be answered from a subset of the annotated turns.

On LongMemEval-S, the MERA-RL run analyzed here reduces the incomplete-recall rate from 37.3% for MERA-Small to 1.1%, while increasing average rounds from 2.88 to 4.16. The shift is downward but to the right: the stronger result combines effective early retrieval (a mean first-hit round of 1.13) with continued search for missing evidence, rather than obtaining completeness by stopping earlier. MERA-RL terminates through retrieval exhaustion in 80.4% of cases, reaches the maximum iteration limit in 8.7%, and stops through verifier-declared convergence in 10.9%. Compared with MERA-Large, it achieves lower incomplete recall (1.1% versus 6.5%) with slightly fewer rounds (4.16 versus 4.36), and therefore dominates MERA-Large on these two diagnostic axes. It nevertheless uses more retrieval rounds than MERA-Small. These results diagnose efficiency and stopping mechanisms rather than constitute a controlled stopping-policy ablation.

Table 6: Online runtime on LongMemEval-S. Calls and wall-clock time are averaged per question and then across three MERA-Small runs, two MERA-RL runs, and three MERA-Large runs. Relative runtime is normalized by MERA-Small.

Configuration	LLM calls/Q	Runtime/Q (s)	Relative runtime
MERA-Small	6.8	43.2	1.00×
MERA-RL	6.6	48.3	1.12×
MERA-Large	10.9	155.5	3.60×

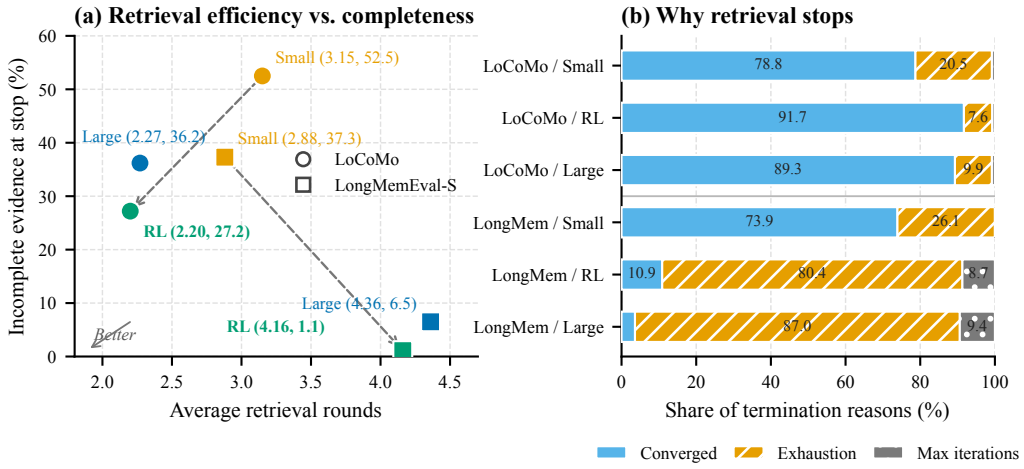

Figure 4: Termination efficiency and stopping mechanisms. Left: average retrieval rounds versus the fraction of questions terminating before all annotated gold turns are recovered; lower-left is better. Within each dataset, the dashed arrow from MERA-Small to MERA-RL shows the change in overall system behavior after reinforcement learning. Right: the complete distribution of convergence, retrieval exhaustion, and maximum-iteration stops. Recall below one is an upper-bound diagnostic because partial gold evidence may already suffice for some answers.

D Planner Behavior and Evidence Packaging

D.1 Retrieval Depth and Retriever Compatibility

Evidence gain diminishes with retrieval depth. At least 92% of each LoCoMo curve's post-first-round recall improvement is realized by round 5, after which the curves change by at most 1.9 points. High-similarity evidence is recovered first, leaving progressively weaker matches and more indirect relations. This diminishing evidence gain motivates adaptive termination rather than mechanically spending the maximum retrieval budget.

Planner outputs must match the retriever. Under dense retrieval on LoCoMo, MERA-RL exceeds MERA-Small by 18.2 points in the first round and by 29.1 points at termination. MERA-Small gains only 2.3 points after round 3, whereas MERA-RL gains a further 6.1 points, indicating that policy training improves both initial query quality and the ability to use accumulated evidence for follow-up retrieval. The earlier keyword diagnostic in Table 7 shows a much smaller RL–Small gap, indicating that planner gains depend on how well the learned query policy matches the retriever.

D.2 Graph-Structured Versus Flat Evidence

Configuration. The no-graph ablation uses dense atomic-fact retrieval with MMR ($\lambda = 0.95$), $k = 8$, and a maximum of 10 retrieval rounds. Parent-turn fact aggregation remains enabled. Qwen3-30B-A3B-Instruct-2507 performs evidence verification and final answer generation. The planners are Qwen3-0.6B without retrieval-grounded training, retrieval-trained Qwen3-0.6B, and

Table 7: LoCoMo query behavior and retrieval outcomes before and after retrieval-grounded training. The comparison between MERA-Small and MERA-RL examines how policy training changes the use of the retrieval interface: query word count and intents per call characterize action construction; repetition and parsing failures characterize action quality; and evidence recall and premature stopping measure downstream retrieval effectiveness. Change is computed as MERA-RL minus MERA-Small.

Metric	MERA-Small	MERA-RL	Change
Average query word count	3.30	3.50	+0.20
Intents per planner call	2.75	2.98	+0.23
Normalized query repetition	17.1%	16.0%	−1.1 pp
JSON parsing failures	0.0%	0.0%	0.0 pp
First-round evidence recall	0.623	0.631	+0.8 pp
Final evidence recall	0.761	0.776	+1.5 pp
Premature stopping	29.0%	27.2%	−1.8 pp

Qwen3-30B without retrieval-grounded training. Figure 3 reports all six no-graph results alongside their graph references on the same question sets used elsewhere in the paper.

Intervention. The flat-evidence variant removes the evidence graph and its associated construction, update, and state-maintenance mechanisms. Accepted evidence accumulates across rounds and remains available to the planner and final answer model as a flat list.

Results. All six graph–flat gaps are positive, with graph accuracies taken from the corresponding configurations in Table 2. For MERA-Small, MERA-RL, and MERA-Large, the gaps are 14.02, 14.41, and 9.65 points on LoCoMo and 5.34, 16.17, and 7.26 points on LongMemEval-S, respectively.

D.3 Changes in Query Behavior

Both planners produce parseable output, so training does not merely repair a formatting failure. Query length increases by only 0.20 words, while the number of intents rises modestly and normalized repetition falls by 1.1 points. Final evidence recall and premature stopping improve in the same direction. These changes are consistent with the incremental reward: repeated evidence receives no additional credit, whereas complementary retrieval actions receive immediate feedback. The present statistics do not isolate the independent contributions of intent count, keyword choice, and stopping behavior.

D.4 A Trajectory from the RL-Trained Planner

We illustrate how accumulated evidence informs what to retrieve next using a recorded LongMemEval-S test trajectory (question ID: `gpt4_5501fe77`). The planner is a GRPO-trained Qwen3-0.6B model. The question is: *"Which social media platform did I gain the most followers on over the past month?"* Neither Twitter, Facebook, nor TikTok is named in the question. We show consecutive rounds 2–4 below, with their preceding evidence state and the eventual answer. Quoted queries and fact excerpts retain the wording in the log; graph descriptions summarize the relevant nodes and relations.

Evidence available after round 1. The initial graph contains only the generic anchors *social media platform*, *followers growth*, and *past month*. The first retrieval batch supplies two useful comparisons. `Evidence_003`, dated May 29, 2023, states: "The user observed that their Twitter follower count increased from 420 to 540 over the past month." `Evidence_004`, dated May 30, states: "The user has observed that their Facebook follower count has remained stable at approximately 800." The verifier adds Twitter and Facebook as active entities, with relations to attributes recording 120 followers gained and a stable follower count, respectively. TikTok is mentioned incidentally in some retrieved records, but has no node or accepted growth evidence in the graph at this stage.

Round 2: querying verified anchors. The planner now issues "What is the number of followers on Twitter over the past month?" and "What is the number of followers on Facebook over the past month?" It also retains a broader comparison query: "Which social media platform had the highest follower increase during the past month?" The named queries use entities established in the preceding graph. This round adds no new platform entity, and TikTok remains absent from the graph.

Round 3: retrieving beyond the current graph. The next batch contains "How many followers did Twitter gain in the past month?", "How many followers did Facebook gain in the past month?", and "Which social media platform had the highest follower increase during the past month?" Retrieval returns `Evidence_025`, dated May 29: "The user gained approximately 200 followers on TikTok over the past three weeks." It also returns `Evidence_028`, dated May 30: "The user has observed a rapid increase in their TikTok follower count." The verifier accepts both records, creates an active TikTok entity, and links it through the `followers_growth_in_period` relation to the newly supported growth attributes. The quantitative evidence is accessible even though TikTok is not yet a node in the graph supplied to this round's planner. The log records the merged retrieval batch, so we do not attribute either record to an individual intent.

Round 4: using the newly admitted entity. The next planner call includes "What is the number of followers on TikTok over the past month?" alongside corresponding queries for Twitter and Facebook. TikTok has thus moved from a retrieved fact into the verified graph and then into a subsequent retrieval intent. Round 4 adds no new accepted evidence IDs; the verifier revisits the existing growth records. This step illustrates how the graph conditions the next action, without implying that every such action yields additional evidence.

Answer and interpretation. The full run ends after six rounds with the logged termination reason `exhaustion`, and returns "TikTok", matching the benchmark reference. This selected trajectory makes the mechanism concrete: verified graph entities provide query anchors, global retrieval admits relevant evidence about an entity outside the current graph, and graph updates make that entity available to subsequent planning. It illustrates the behavior of an RL-trained planner; it does not by itself isolate the contribution of RL or establish an efficiency advantage.

D.5 Parent-Turn Fact Aggregation

MERA performs matching over atomic facts and groups each match with all other facts extracted from the same parent turn before verification. This design aims to combine fine-grained matching with additional facts associated with the retrieved turn. We ablate parent-turn fact aggregation on all 1,986 LoCoMo questions: each matched atomic fact is returned directly as a separate candidate. Both variants retain speaker and timestamp metadata. Within each planner row, the memory cache, code version, retrieval parameters, answer model, and judge are held fixed. The analysis therefore concerns the paired change within each row rather than absolute accuracy differences across planners.

Removing parent-turn fact aggregation changes answer accuracy by -0.46 points for MERA-RL, -2.77 points for MERA-Small, and -4.66 points for MERA-Large. All three paired comparisons favor aggregation, with larger drops for the two planners without retrieval-grounded training. These results are consistent with sibling facts providing useful information beyond the matched fact. The comparison does not test access to the original wording of the turn, since both variants present extracted facts.

One plausible explanation for MERA-RL's smaller drop is that retrieval-grounded training encourages queries whose matched facts more directly address the question, reducing reliance on additional facts from the same parent turn. The reward favors newly recovered evidence, and Table 7 reports slightly more intents, less repetition, and higher final evidence recall after training. These observations are compatible with more targeted retrieval, but do not directly measure the sufficiency of individual matched facts or establish why MERA-RL is less sensitive to aggregation.

The ablation also changes how evidence is packaged. Without aggregation, the verifier receives shorter fact records, while sibling facts from one source turn can occupy separate candidate slots

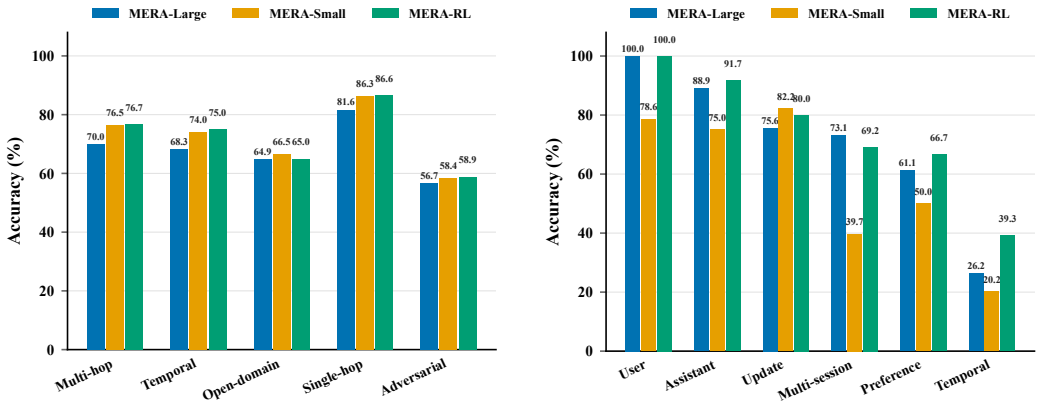

Figure 5: Accuracy by question category on (left) LoCoMo and (right) LongMemEval-S. Values above bars are accuracies (%).

despite a fixed candidate budget. The result therefore evaluates the aggregation interface as a whole; it does not isolate the contributions of additional fact content, input length, or candidate competition.

E Question Types and Failure Attribution

E.1 Accuracy by Question Type

LoCoMo shows a consistent category ordering across planners: single-hop accuracy is highest, followed by multi-hop, temporal, open-domain, and adversarial accuracy. The high single-hop scores are consistent with questions that can be resolved from a localized fact. Adversarial questions test whether a model avoids unsupported answers, so their lower accuracy should be interpreted as difficulty handling absent information rather than updating stored knowledge. Open-domain questions can require knowledge beyond the conversation, making their scores less directly attributable to conversational evidence retrieval alone. Differences between MERA-Small and MERA-RL remain within 1.5 points for every category, consistent with the small overall improvement in the matched keyword setting.

The largest descriptive difference between MERA-Small and MERA-RL on LongMemEval-S occurs for multi-session questions (29.5 points), followed by single-session user information (21.4 points) and temporal questions (19.1 points). Multi-session questions naturally expose decomposable information needs, allowing evidence from one session to anchor retrieval in another. Temporal accuracy remains only 39.3%, however, showing that making temporal evidence visible does not guarantee correct normalization, ordering, or arithmetic.

E.2 Failure Attribution

Long-term memory QA can be approximated as

$$P(\text{correct}) \approx P(\text{evidence sufficient})P(\text{correct} \mid \text{evidence sufficient}). \tag{10}$$

The attribution analysis covers 655 incorrect LoCoMo answers and 31 incorrect LongMemEval-S answers. Reasoning errors are the largest category on both datasets, accounting for 63.5% of LoCoMo failures and 87.1% of LongMemEval-S failures. This category provides a coarse stage-level diagnosis: it shows that the system found supporting evidence but still produced an incorrect answer. Identifying the root cause requires a second-level breakdown by question and error type.

The LongMemEval-S breakdown exposes three distinct causes. First, temporal-reasoning questions contribute 40–60% of reasoning errors across configurations; in the MERA-RL run analyzed here, they account for 14 of 27 cases (51.9%). Inspection attributes many of these failures to incorrect

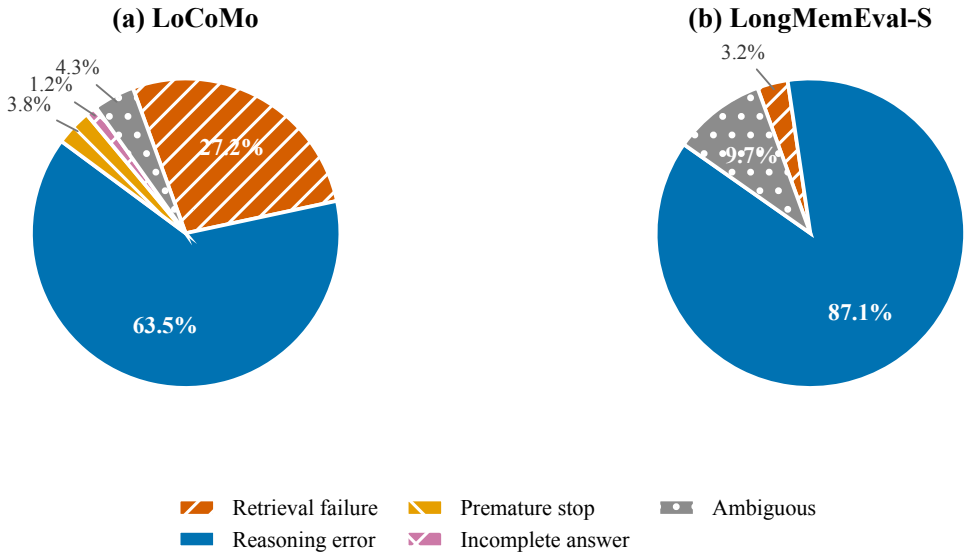

Figure 6: Attribution of incorrect MERA-RL answers on LoCoMo and LongMemEval-S. "Reasoning error" denotes an incorrect final answer despite recovered evidence; it localizes the failed stage but is not itself a fine-grained root cause. Percentages are diagnostic labels over errors rather than manually adjudicated absolute rates.

relative-date calculations, so evidence availability alone does not solve temporal normalization and arithmetic. Second, single-session-user questions contribute no reasoning errors: once the relevant simple fact is retrieved, the answer model responds correctly. Third, multi-session questions contribute a nearly unchanged 7–9 reasoning errors across planner configurations, indicating that cross-session evidence integration remains difficult even as retrieval improves. Reasoning error is therefore an intermediate diagnosis, while temporal calculation and cross-session integration are the more specific downstream root causes.

The balance of retrieval and reasoning errors differs sharply across datasets. Retrieval failure accounts for 27.2% of LoCoMo errors but only 3.2% on LongMemEval-S, consistent with the lower final evidence recall on LoCoMo in Figure 3. A LoCoMo question has approximately three to four gold turns on average, compared with about 1.5 on LongMemEval-S, making complete evidence acquisition more difficult. By contrast, stronger retrieval on LongMemEval-S shifts most remaining errors to downstream reasoning. Premature stopping is uncommon in both runs (3.8% and 0.0%), so it is not the primary source of failure in this comparison.

The attribution combines deterministic trajectory checks with local 30B-model judgments. The percentages in Figure 6 should therefore be interpreted as diagnostic evidence about bottlenecks rather than exact causal rates.

F Retrieval Hyperparameter Sensitivity

With $k = 8$ and $T_{\max} = 10$, accuracy on LoCoMo is 75.73%. Setting $k = 4$ or $k = 16$ yields 74.97% and 77.63%, respectively, while reducing the retrieval budget to five rounds yields 75.88%. Across these settings, accuracy ranges from 74.97% to 77.63%. Increasing retrieval depth to $k = 16$ gives the highest observed accuracy, whereas reducing the round budget to five yields accuracy close to the default setting. The latter observation is consistent with the average retrieval length of approximately 2.2 rounds, as most questions terminate before reaching either budget.