REMIND: Evidence-Conditioned Memory Invocation over Long-Term Conversations via Agentic Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

Long-term conversation memory is fundamental for Large Language Models (LLMs) to maintain coherent personalization and deliver contextually grounded responses across extended interactions. Recent work addresses this through retrieval-augmented generation, retrieving only question-relevant information from the memory bank to avoid hallucinations caused by irrelevant contextual noise. *However, the most critical memories for answering a question are often not the most semantically similar ones.* Memory relevance also arises from causal relations and temporal recency. A semantically distant memory can capture the root cause of a situation, whereas a highly similar one may reflect an obsolete state. Based on this novel insight, we present **REMIND** [1], a reinforcement-learning framework for selective, on-demand access to personal memory. Specifically, REMIND interleaves two processes on demand. *Evocation* constructs adaptive cues from the question and accumulated evidence, and *Invocation* accesses typed memory banks and inspects the returned evidence to decide whether to recall further or answer. Across 2,386 questions from LoCoMo, LongMemEval, and MemoryAgentBench, REMIND with a Qwen3-8B backbone achieves relative improvements of 12.5% over Base RAG and 4.8% over Search-R1 with same backbone. This gain highlights the value of adaptive, evidence-conditioned recall over fixed retrieval.

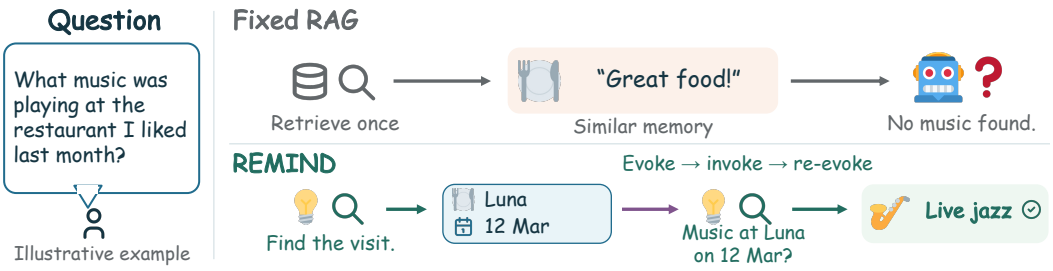


Figure 1: Illustrative restaurant-memory question. Fixed RAG retrieves a similar comment about food but no music. REMIND first identifies the restaurant and visit date, then uses that evidence to evoke a more specific cue and retrieve the music detail.

1 Introduction

As language models become persistent personal assistants, they must connect new questions with interactions spanning months or years. Long-term memory maintains useful information beyond the immediate context, including a stable user profile, past events, learned facts, recurring procedures, and preferences. Existing systems increasingly organize such information through context tiers, separate memory readers, graphs, or linked notes (Packer et al., 2023; Wang et al., 2023b; Gutiérrez et al.,

[1] Our code is released on https://anonymous.4open.science/r/remind_code-D6B5

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.

2024; Xu et al., 2025). These designs have made memory construction and maintenance substantially more capable.

Yet constructing a useful store does not determine how to access it for a new question. Memory construction decides what to write, update, or connect across sessions. Answer-time access must formulate an appropriate cue, identify relevant information, and judge whether the evidence is sufficient. A well-organized fact may remain inaccessible when the current question differs from its stored wording. Holding the memory substrate fixed lets us isolate this access policy. A question such as "What happened to that project after our last meeting?" may first require resolving the project and meeting before its later status. The challenge is reconstructing the cue, not rewriting the memory.

Answer-time access generally follows static one-shot retrieval or dynamic search. Fixed retrieval-augmented generation (RAG) queries once with the original question and prepends a fixed number of memories (Lewis et al., 2020), fixing the cue, evidence depth, and stopping point before inspecting memory. Other methods learn query rewriting (Ma et al., 2023; Wang et al., 2025), control its steps (Ye et al., 2025), or retrieve iteratively (Trivedi et al., 2023; Jiang et al., 2023; Asai et al., 2024; Jeong et al., 2024). Agentic-search methods further optimize repeated retrieval with reinforcement learning (Jin et al., 2025; Chen et al., 2025).

Long-term conversational memory adds two requirements that these access paradigms do not jointly address:

- **The memory that answers a question is often not the one most similar to it.** Personalized relevance is *relational*: it arises from *causal*, *temporal*, *contrastive*, *conditional*, and even *negation* relations rather than simple similarity. A memory that best matches the question wording may describe the wrong time period or an obsolete state, while the decisive memory shares little lexical overlap. Simple retrieval, keyed on the original question, cannot revise its cue once an intermediate person, event, or relation is resolved.
- **Memory activation depends on recalled evidence.** External-corpus agents reformulate queries to locate public information. Personal recall often proceeds associatively: one episode supplies a person, date, or relation that becomes the cue for the next. Cue formulation, memory access, evidence inspection, and stopping are thus interdependent. Final-answer supervision alone does not directly identify which intermediate decisions found new evidence or merely repeated it.

Personal memory access introduces a different information environment. Conversational memories are user-specific, often compressed from prior dialogue, and frequently elliptical: one memory may identify a person or event without containing the information ultimately required by the current question. They are also heterogeneous. Episodic memories describe particular events, semantic memories encode user-specific facts, procedural memories capture recurring preferences or practices, and core memories summarize stable aspects of the user. Moreover, personal facts can change over time, so a semantically close memory may describe the correct entity but the wrong state or time period. The policy must therefore reason not only about *what to search for*, but also about *which kind of memory can fill the current information gap*, *whether the recovered evidence remains temporally valid*, and *whether further recall is warranted*.

We propose **REMIND**, reinforcement learning for memory invocation on demand. It adapts agentic retrieval to long-term conversational memory, where relevant evidence may be linked to the question through causal or temporal relations rather than direct similarity. Each invoked memory can evoke the next cue, allowing the policy to follow these relations through multi-hop recall until the accumulated evidence is sufficient to answer. Evocation, invocation, and stopping are learned as one evidence-conditioned policy. REMIND decomposes selective memory recall into two interleaved operations. *Evocation* formulates or refines a retrieval cue from the question and the evidence accumulated so far. *Invocation* uses that cue to access personal memory and inspect the returned evidence. Newly recovered entities, events, or relations can support a more precise subsequent cue, while sufficient or redundant evidence informs the decision to stop. This evidence-conditioned loop enables the model to follow temporal and logical connections across memories rather than treating retrieved items as an unordered collection of semantically similar text.

In REMIND-GRPO, final QA provides an outcome signal, while memory-specific process signals cover evidence-session coverage, valid memory actions, cue grounding, first-time evidence gain, and redundant-recall penalties. These signals connect answer quality to the decisions that produced it

without requiring a reward model at inference time. We evaluate the resulting policy against fixed RAG under matched backbones and a shared memory substrate.

Figure 1 summarizes this contrast: fixed RAG follows a one-shot path, whereas REMIND learns an adaptive access trajectory motivated by the staged organization of human autobiographical recall. Recall begins with a *construction* stage that searches for and selects a specific episode. A later *elaboration* stage recovers and reassembles its details (Audrain et al., 2022; Daviddi et al., 2023). REMIND adopts this staged organization as a computational abstraction. We call cue construction or refinement *evocation*. We call goal-directed memory access and evidence elaboration *invocation*. Each invocation can re-evoke a sharper cue from newly recovered evidence or signal that recall should stop. Through this evidence-conditioned loop, the policy can track temporal order and causal links across recalled events instead of treating each memory in isolation.

Our contributions are threefold.

- We translate the construction and elaboration account of autobiographical recall into an evocation and invocation policy over a fixed personal memory store, separating memory access from both memory construction and generic external-corpus search.
- We introduce REMIND-GRPO, which adapts group-relative optimization to memory-access trajectories through supervision for cue grounding, evidence coverage, new evidence, redundant recall, valid actions, and answer quality.
- Across a unified evaluation set assembled from LoCoMo, LongMemEval, and MemoryAgentBench, Qwen3-8B REMIND achieves relative improvements of 12.5% over same-backbone Base RAG and 4.8% over the reported Search-R1 reference.

2 Related Work

Retrieval and long-term memory. Retrieval-augmented models learn dense retrieval, passage fusion, or retrieval during language modeling (Guu et al., 2020; Karpukhin et al., 2020; Izacard et al., 2022; Izacard & Grave, 2021). Long-term memory systems use context tiers, summaries, separate readers, graphs, linked notes, and resource managers (Packer et al., 2023; Wang et al., 2023b; Gutiérrez et al., 2024; Xu et al., 2025; Kang et al., 2025; Chhikara et al., 2025). Recent systems further strengthen graph-based access (Gutiérrez et al., 2025). Memory benchmarks cover long conversations, long contexts, and distinct abilities (Maharana et al., 2024; Wu et al., 2025; Bai et al., 2024; Hu et al., 2026). REMIND studies how a policy accesses an existing memory substrate for each question.

Learned memory management. Memory-R1 learns structured operations with separate manager and answer agents. AgeMem and UMA integrate a broader memory lifecycle into an agent policy (Yan et al., 2026; Yu et al., 2026b; Zhang et al., 2026). CMI-Mem Wang et al. (2026) trains a typed manager. DeferMem (Yin & Tang, 2026) learns query-conditioned evidence distillation after broad retrieval. APEX-MEM (Banerjee et al., 2026) and TA-Mem (Yuan et al., 2026) instead pair newly constructed structured stores with multi-tool or iterative retrieval agents at query time. REMIND keeps construction and the retriever fixed, and trains the query-time access and answer trajectory.

Query rewriting and adaptive retrieval. Query reformulation can draw on document recall or generated pseudo-documents. It can also use feedback from the retriever, reader, reranker, or final answer (Nogueira & Cho, 2017; Gao et al., 2023; Wang et al., 2023a; Wu et al., 2022; Ma et al., 2023; Mao et al., 2024; Wang et al., 2025). Q-PRM learns variable-length rewriting (Ye et al., 2025). Other systems retrieve during reasoning, reflect, route retrieval, or correct results (Trivedi et al., 2023; Jiang et al., 2023; Asai et al., 2024; Jeong et al., 2024; Su et al., 2024; Yan et al., 2024; Wang et al., 2024). REMIND unifies reformulation, repeated memory access, stopping, and answering.

Reinforcement learning for search. ReAct (Yao et al., 2023), Toolformer Schick et al. (2023), and WebGPT Nakano et al. (2021) study language-model tool use. GRPO avoids a value model. Search-R1 Jin et al. (2025), ReSearch Chen et al. (2025), and R1-Searcher Song et al. (2025) train the LLM for repeated searches. These search-RL methods optimize information seeking over external corpora. MAPD instead combines agentic search with distillation by converting proprietary-model

traces into a structured, style-normalized protocol for joint distillation and RL (Liu et al., 2026). REMIND trains a policy for relational, on-demand recall over a personal memory store that records evolving user states. Recalled evidence shapes the next cue, while memory-specific rewards promote evidence coverage and suppress redundant retrieval.

Adjacent trainable access policies. Reflective Memory Management combines prospective multi-granularity memory construction with online-RL retrieval refinement based on cited evidence (Tan et al., 2025). MemSearcher jointly trains reasoning, search, and compact memory updates for multi-turn search agents (Yuan et al., 2025). LazyMem defers query-conditioned memory construction until after broad retrieval, and MemChain learns grounded post-retrieval evidence traces (Yu et al., 2026a; Ma et al., 2026). These systems strengthen query-time memory processing through different interfaces. REMIND keeps an existing typed personal-memory substrate and retriever fixed while training repeated cue generation, invocation, stopping, and answering.

3 Method

REMIND holds memory writing, summarization, and organization fixed and learns an access-and-answer trajectory over the resulting store. The policy conditions each query and its stopping decision on evidence from earlier turns. We first formalize this interaction and then describe the policy, retriever, answer path, and learning objective.

3.1 Problem Formulation

Let x be the current question. The user's long-term memory is $\mathcal{M} = \{\mathcal{M}^c, \mathcal{M}^e, \mathcal{M}^s, \mathcal{M}^p\}$. Its four banks contain core, episodic, semantic, and procedural memories. The task is to produce an answer y supported by this memory. Table 4 in Appendix B collects the notation.

Fixed RAG applies one retrieval function, $\mathcal{R}(x, \mathcal{M}, K)$, and answers from that single result; K is its fixed retrieval depth. REMIND instead learns a policy π_θ with parameters θ. Its initial history h_1 contains x, summaries of the four banks, and optional temporal context. Before turn t, $h_t = (h_1, a_1, o_1, \ldots, a_{t-1}, o_{t-1})$ also contains all preceding actions and observations. The policy produces the trajectory in Equation 1:

$$\tau = (x, a_1, o_1, \ldots, a_T, o_T, y), \tag{1}$$

where a_t is the action selected from h_t and o_t is the environment response. T is the number of tool actions before answer y. The memory banks stay fixed during this process. Only the query and the gathered evidence change from turn to turn.

3.2 Evoke and Invoke Policy

Figure 2 gives an overview. Research on autobiographical recall links memory construction to the anterior hippocampus and distinguishes construction from later elaboration (Audrain et al., 2022; Daviddi et al., 2023). REMIND uses these stages as computational roles. Evocation constructs the retrieval cue. Invocation accesses memory and uses the recovered evidence to decide whether to invoke memory again or answer. The policy observes the question, compact summaries of the four memory banks, optional temporal context, and all previous tool responses. Invoked evidence can therefore trigger a new evocation. Because the history retains earlier observations, the next cue can follow temporal order or causal and contrastive links expressed across recalled events. A memory invocation tests this cue against one user's fixed conversational history, including typed memories and linked source dialogue.

Evocation is the cue-construction stage of answer-time access. Given the current history h_t, the policy converts the conversational question x into a retrieval-oriented query q_t. The first cue need not repeat the question. A conversational utterance may use pronouns, generic question words, or elliptical phrases such as "that project" and "our last meeting." Such expressions are clear in context but often poor keys for a fixed memory store. The policy keeps the entities and relations that distinguish the target memory, removes wording that does not help retrieval, and makes implicit constraints explicit. These constraints can involve time, cause, contrast, preference, or relations across sessions.

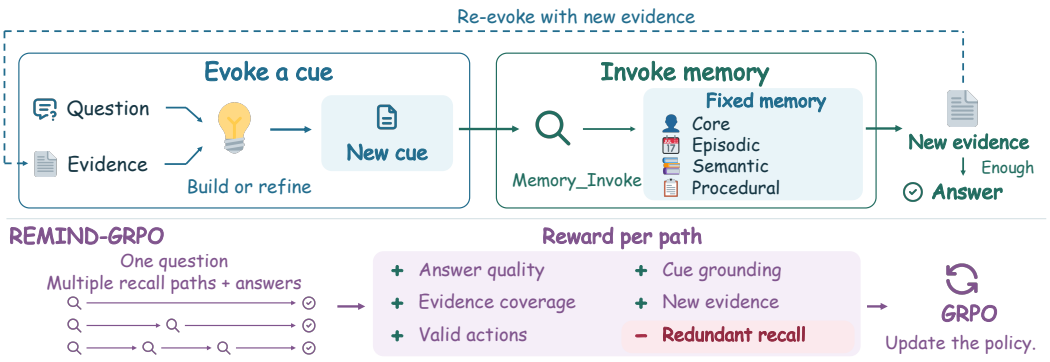


Figure 2: REMIND access and training pipeline. The policy builds a cue from the question and prior evidence, invokes the fixed four-bank memory store, and either re-evokes with new evidence or answers. For each question, REMIND-GRPO scores sampled recall paths and answers using answer quality, evidence coverage, valid actions, cue grounding, and first-time evidence gain, with a penalty for redundant recall.

Equation 2 makes the evidence-conditioned state update explicit:

$$h_{t+1} = (h_t, a_t, o_t), \qquad a_{t+1} \sim \pi_\theta(\cdot \mid h_{t+1}). \tag{2}$$

Here h_t is the history before turn t, a_t is its memory action, o_t is the returned evidence, and π_θ is the learned policy. The updated history h_{t+1} includes that evidence. If a_{t+1} invokes memory again, its cue q_{t+1} can use a newly recovered person, date, or event. For "What happened to that project after our last meeting?", a first invocation can identify the meeting date, which then sharpens the cue for the project's later status. The next cue can thus follow a temporal, causal, or contrastive relation found in an intermediate memory.

Invocation is the subsequent interaction with memory. The policy action a_t has one of two XML-formatted forms in Equation 3:

$$a_t \in \begin{cases} \texttt{Memory_Invoke}(q_t, K_t), \\ \texttt{Memory_Finalize}(r_t) \end{cases}. \tag{3}$$

`Memory_Invoke` uses cue q_t and requests K_t memories at turn t. The policy generates both fields; K_t may differ from the prompt default. Its observation o_t contains the returned memories and source dialogue when available. `Memory_Finalize` records a short rationale r_t, ends the invocation phase, and moves the policy to answering.

The policy may finalize immediately when no memory is needed. A direct memory question may finish after one invocation, while a multi-session or temporal question may require further recall. In that case, the policy can use a name, date, or event found in o_t to write a more specific q_{t+1}. The implementation allows at most five memory-invocation calls. An unparsable action, free text before finalization, or exhaustion of this budget makes the trajectory invalid. Algorithm 1 in Appendix A summarizes this control flow.

3.3 Hybrid Retrieval and Answering

Each `Memory_Invoke` call retrieves from the union of the four typed banks. The policy currently accesses all four banks rather than selecting a bank in its action. This keeps the learned decision focused on query content, retrieval depth, repeated access, and stopping.

For memory entry m and query q, let $r_{\mathrm{BM25}}(m)$ and $r_{\mathrm{emb}}(m)$ be its zero-based lexical and embedding ranks. We fuse them as in Equation 4:

$$s(m \mid q) = \frac{3}{60 + r_{\mathrm{BM25}}(m)} + \frac{7}{60 + r_{\mathrm{emb}}(m)}, \tag{4}$$

Here $s(m \mid q)$ is the fused score, 60 is the rank offset, and a missing item has infinite rank. The 3:7 BM25-to-embedding weights follow the 0.3/0.7 sparse-to-dense ratio used by OpenClaw.[2] Reciprocal ranks avoid calibrating the two score scales. The retriever returns the highest-scoring K_t entries, attaches source dialogue when available, and removes exact duplicate chunks across turns.

The identifiers of the evidence sessions returned at turn t form S_t. The full response is appended to the policy history as o_t. The policy can therefore compare new evidence with earlier evidence before invoking memory again or stopping. Tool-response tokens are environment output and are masked from the RL loss. Gradients apply only to policy-generated invocation, finalization, and answer tokens.

After `Memory_Finalize`, the same policy receives an answer instruction and generates y from the collected evidence. Thus, Qwen3-4B is normally both the memory-control policy and the answer model. Qwen3.7-Max is used for the standalone RAG baseline and evaluation judge, not as REMIND's retrieval controller.

Training and inference. The training pool is constructed by processing source data from LoCoMo and LongMemEval; MemoryAgentBench is used only in evaluation. During training, the policy samples complete trajectories and receives the reward defined in the next subsection. During inference, it runs the same search and finalization loop without a reward or judge signal. The policy answers using only the question, memory summaries, and evidence that it has collected. This keeps the evaluation judge outside the inference loop.

3.4 REMIND-GRPO: Group-Relative Optimization with Multi-Component Reward

We train the trajectory policy with **REMIND-GRPO**. It retains the group-relative objective of GRPO (Shao et al., 2024) and defines credit at the level of a complete memory-access trajectory through the multi-component reward below. For each question, GRPO samples a group of G trajectories and compares their rewards within the group. Trajectories that answer and access memory more successfully therefore receive larger relative advantages without a separate value network.

REMIND-GRPO scores final answers (R_{qa}), retrieval (R_{ret}), valid termination (R_{fmt}), and cues (R_{evoke}). First-time gold evidence g_{new} and redundant-query count N_{dup} distinguish productive follow-up invocations from repeated ones.

For trajectory i, let $\widehat{A}_i$ be its advantage relative to the sampled group. Let $z_{i,j}$ be its j-th policy-generated token and $z_{i,<j}$ the preceding context. The rollout policy π_{old} generated the group; $\rho_{i,j} = \pi_\theta(z_{i,j} \mid z_{i,<j})/\pi_{\text{old}}(z_{i,j} \mid z_{i,<j})$ is the token probability ratio. GRPO clips it with the surrogate in Equation 5:

$$\ell_{i,j} = \min\left\{\rho_{i,j}\widehat{A}_i,\ \text{clip}(\rho_{i,j}, 1-\epsilon, 1+\epsilon)\widehat{A}_i\right\}, \tag{5}$$

Here $\ell_{i,j}$ is the clipped surrogate for token j in trajectory i, and ϵ is its clipping range. Let $\mathcal{Z}_i$ be the set of trainable token positions in that trajectory and G the number of sampled trajectories for the question. Equation 6 averages the surrogate over these tokens and trajectories and adds a KL regularizer:

$$J(\theta) = \mathrm{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathcal{Z}_i|}\sum_{j\in\mathcal{Z}_i}\ell_{i,j} - \beta D_{\text{KL}}(\pi_\theta\|\pi_{\text{ref}})\right]. \tag{6}$$

Here $J(\theta)$ is the training objective, E averages over training questions and sampled groups, D_{KL} is KL divergence, β weights it, and π_{ref} is the fixed reference policy. Tool observations remain in the policy context but are excluded from $\mathcal{Z}_i$ by the loss mask.

The base reward checks three requirements of a useful trajectory. It should answer correctly, retrieve annotated evidence sessions, and terminate with a valid tool sequence. Equation 7 combines these signals:

$$R_{\text{base}} = 0.5R_{\text{qa}} + 0.3R_{\text{ret}} + 0.2R_{\text{fmt}}. \tag{7}$$

[2]OpenClaw official repository: https://github.com/openclaw/openclaw.

$R_{\text{qa}} \in \{0, 1\}$ judges the final answer during training, $R_{\text{fmt}} \in \{0, 1\}$ indicates valid `Memory_Finalize` termination, and R_{ret} is retrieval F1. Equation 8 defines the latter from per-entry precision and gold-session recall:

$$\begin{aligned} p_{\text{ret}} &= \frac{1}{L}\sum_{\ell=1}^{L}\mathbf{1}[C_\ell \cap \mathcal{G} \neq \emptyset], \\ r_{\text{ret}} &= \frac{\left|\mathcal{G} \cap \bigcup_{\ell=1}^{L} C_\ell\right|}{|\mathcal{G}|}, \\ R_{\text{ret}} &= \frac{2p_{\text{ret}}r_{\text{ret}}}{p_{\text{ret}} + r_{\text{ret}}}. \end{aligned} \tag{8}$$

Here L counts all entries returned across the trajectory, C_ℓ is the candidate source-session set for entry ℓ, and $\mathcal{G}$ is the annotated gold-session set. An entry contributes to p_{ret} if C_ℓ overlaps $\mathcal{G}$; unmapped entries still count in L. The recall r_{ret} counts distinct gold sessions covered. We set $R_{\text{ret}} = 0$ when $L = 0$, $\mathcal{G} = \emptyset$, or $p_{\text{ret}} + r_{\text{ret}} = 0$. With no annotated gold session, the R_{ret} term is removed from Equation 7 and the QA and format weights become $5/7$ and $2/7$. The training-time QA judge is distinct from the evaluation judge used for reported scores.

Answer correctness alone does not say which invocation found useful evidence. At invocation turn t, let S_t be the union of candidate sessions linked to its returned entries, and let $C_{t-1} = \bigcup_{j<t} S_j$ contain sessions retrieved earlier. For the gold-session set $\mathcal{G} \neq \emptyset$, the normalized gain g_{new} sums over invocation turns and credits each gold session when it first appears:

$$g_{\text{new}} = \frac{1}{|\mathcal{G}|}\sum_{t} \left|(S_t \cap \mathcal{G}) \setminus C_{t-1}\right|. \tag{9}$$

Thus, each gold session contributes only upon its first retrieval; when $\mathcal{G} = \emptyset$, we set $g_{\text{new}} = 0$.

Let $\mathcal{I}_\tau \subseteq \{1, \ldots, T\}$ index the `Memory_Invoke` actions. At each $t \in \mathcal{I}_\tau$, a judge assigns $j_t^{\text{rel}}, j_t^{\text{qual}} \in \{0, \ldots, 10\}$ for the relevance of q_t to x and its rewrite quality (faithfulness, specificity, completeness, and conciseness). With fixed $\alpha = 0.6$, the normalized score is

$$s_t = \frac{\text{round}\left(\alpha j_t^{\text{rel}} + (1-\alpha) j_t^{\text{qual}}\right)}{10}. \tag{10}$$

For trajectory τ, the cue-grounding reward averages per-query scores s_t over its invocation-turn set $\mathcal{I}_\tau$, or is zero if the set is empty:

$$R_{\text{evoke}}(\tau) = \begin{cases} |\mathcal{I}_\tau|^{-1}\sum_{t\in\mathcal{I}_\tau} s_t, & |\mathcal{I}_\tau| > 0, \\ 0, & |\mathcal{I}_\tau| = 0. \end{cases} \tag{11}$$

For trajectory τ, Equation 12 combines the base answer, retrieval, and format score R_{base} with the cue score R_{evoke}, first-time gold-evidence gain g_{new}, and redundant-query count N_{dup}:

$$\begin{aligned} R(\tau) \quad = \quad & (1 - w_g)R_{\text{base}} + w_g R_{\text{evoke}}(\tau) \\ & + \lambda_{\text{gain}} g_{\text{new}} - \lambda_{\text{dup}} N_{\text{dup}}. \end{aligned} \tag{12}$$

$R(\tau)$ is the complete trajectory reward. N_{dup} counts follow-up queries whose token Jaccard similarity with an earlier query is at least 0.5 and that retrieve no new gold session. The coefficients w_g, λ_{gain}, and λ_{dup} weight cue grounding, new evidence, and redundant search; the latter two are 0.30 and 0.10. The supplied Search-R1 reference has $w_g = 0$, and the REMIND-GRPO variants use $w_g \in \{0.1, 0.2, 0.3, 0.4\}$. We compare complete configurations as operating points and use $w_g = 0.2$ in the main evaluation because it has the highest overall QA score.

The outcome, coverage, protocol, grounding, gain, and duplicate signals assign credit to the complete evoke–invoke trajectory. None of these reward models is called at inference time.

4 Experiments

We evaluate same-backbone QA gains (RQ1), question-type changes (RQ2), and reported reward and retrieval configurations (RQ3) under one shared protocol.

Backbone	System	Access	Training	Avg. score ↑	Δ_{Base} (pp) ↑
Qwen3-4B	Base RAG	One-shot	None	0.6186	–
	REMIND	Iterative	RL	0.6790	+6.04
Qwen3-8B	Base RAG	One-shot	None	0.6459	–
	Search-R1 ($w_g = 0$)	Iterative	RL	0.6932	+4.73
	REMIND ($w_g = 0.2$)	Iterative	RL	0.7267	**+8.08**
Qwen3.7-Max	Base RAG	One-shot	None	**0.7615**	–

Table 1: Overall QA results and absolute gains over same-backbone Base RAG.

4.1 Experimental Setup

Evaluation data. We evaluate on a single merged set of 2,386 questions: 1,188 from LoCoMo (Maharana et al., 2024), 752 from LongMemEval (Wu et al., 2025), and 446 from MemoryAgentBench (Hu et al., 2026). LoCoMo tests very long multi-session conversations, LongMemEval targets sustained interactive memory, and MemoryAgentBench uses incremental interactions to probe distinct memory abilities. We pool these complementary sources under one judging protocol for the overall comparison and retain a shared seven-type label space for fine-grained analysis. This breakdown is available for the 1,940 LoCoMo and LongMemEval questions. It contains 50 assistant, 27 preference, 902 knowledge-update, 435 temporal, 190 multi-session, 282 single-session, and 54 single-session-user questions. The 446 MemoryAgentBench questions are included in the overall score but not in the type breakdown.

Systems. We use Qwen3-4B and Qwen3-8B to test the access policy at two scales while keeping the model family fixed. *Base RAG* denotes the corresponding unadapted model with one fixed retrieval step and no REMIND training. The 8B $w_g = 0$ row is the supplied Search-R1 reference (Jin et al., 2025). The four REMIND rows are reported with cue-grounding reward weights from 0.1 to 0.4. The main $w_g = 0.2$ setting prompts Top-3 retrieval by default; the policy can request another K_t. We also include Qwen3.7-Max with fixed RAG as a higher-capacity reference. The same-backbone Qwen3 comparisons provide the primary evidence; Qwen3.7-Max indicates the remaining gap to a larger model. By reusing the memory substrate across the Qwen3 systems, the evaluation focuses on answer-time access rather than differences in memory construction.

All reported reinforcement-learning runs used eight NVIDIA H20 GPUs.

Evaluation. Qwen3.7-Max judges each generated answer against the gold answer. It uses the `ACCURACY_PROMPT` in `code/eval/llm_judge.py` with temperature 0.0 and returns a JSON label of `CORRECT` or `WRONG`. These labels map to one and zero. The overall score is the mean over all 2,386 questions. Each type score is a separate mean over that type. It is not used to reconstruct the overall result. Query analysis reports normalized edit distance, ROUGE-L with the original question, number of invocation calls, retrieval F1 as defined in Equation 8, and the same QA score.

4.2 Main Experiments

RQ1. Does REMIND Improve Base RAG? Table 1 shows that REMIND improves both same-backbone Base RAG systems. Qwen3-4B rises from 0.6186 with Base RAG to 0.6790 with REMIND, a gain of 6.04 percentage points. For Qwen3-8B, the highest-scoring REMIND setting reaches 0.7267, whereas Base RAG scores 0.6459. The difference is 8.08 points. These comparisons fix the backbone size, memory substrate, and evaluation protocol. They measure the full access-and-answer system; REMIND may make more retrieval calls than Base RAG.

Qwen3-4B REMIND also exceeds unadapted Qwen3-8B Base RAG by 3.31 points, while Qwen3-8B REMIND narrows the observed gap to the Qwen3.7-Max Base RAG reference.

RQ2. Which Question Types Improve? The type breakdown in Table 2 separates the two model scales. At 8B, REMIND improves every reported type over Base RAG, including multi-session questions from 0.4947 to 0.5684 and temporal questions from 0.6069 to 0.6943. In one trajectory, the cue

System	Assistant	Preference	Knowledge upd.	Temporal	Multi-session	Single-session	Single-user
Qwen3-4B Base RAG	0.7200	**0.5926**	**0.8126**	**0.6506**	0.4421	0.5142	0.8519
REMIND-4B	**0.8400**	0.5556	0.8027	0.6253	**0.5263**	**0.5709**	**0.9444**
Qwen3-8B Base RAG	0.7800	0.4815	0.7550	0.6069	0.4947	0.5213	0.8704
REMIND-8B	**0.9600**	**0.7037**	**0.8426**	**0.6943**	**0.5684**	**0.6099**	**0.9815**

Table 2: Per-type QA results for same-backbone Base RAG and REMIND comparisons with Qwen3-4B and Qwen3-8B over the labeled portion of the merged evaluation set, using its unified LoCoMo–LongMemEval type labels.

w_g	Edit dist.	ROUGE-L	Invokes	Ret. F1 ↑	QA ↑	Δ_{SR1} (pp) ↑
$0^{\dagger}$	–	–	–	–	0.6932	–
0.1	0.6636	0.2777	1.49	0.6192	0.7238	+3.06
0.2	0.6917	0.2576	1.41	0.6307	**0.7267**	**+3.35**
0.3	0.6811	0.2571	1.33	0.6224	0.7121	+1.89
0.4	0.6966	0.2456	1.19	**0.6326**	0.7096	+1.64

Table 3: Query and outcome measurements across 8B configurations. †Search-R1 supplies QA only; process metrics are available for REMIND.

"plants acquired last month gardening new purchases" is refined with "May 2023" after retrieval resolves the time reference. Further query examples appear in the supplementary material.

At 4B, REMIND improves four types while preference, knowledge-update, and temporal scores decline; at 8B, all seven rise. One policy handles cue construction, evidence inspection, stopping, and answering, making scale relevant to the complete trajectory.

The type table covers 1,940 LoCoMo–LongMemEval questions; all 2,386 questions contribute to the overall score, which remains the main comparison.

At 8B, the largest absolute gains are 22.22 points for preference and 18.00 for assistant questions; even the smallest, multi-session, is 7.37 points. At 4B, preference, knowledge-update, and temporal scores instead fall by 3.70, 0.99, and 2.53 points, respectively. The benefit is therefore scale-dependent rather than uniform across question types.

4.3 Reward and Retrieval Configurations

RQ3. How do the reported configurations compare? All four REMIND rows in Table 3 exceed the supplied Search-R1 reference. The $w_g = 0.2$ row has the highest QA (0.7267), whereas $w_g = 0.4$ has the highest retrieval F1. Higher w_g accompanies fewer invocations and lower ROUGE-L across these rows. As complete configurations, they do not isolate the effect of w_g.

At $w_g = 0.2$, prompted defaults $K = 1, 3, 5$ yield QA scores of 0.7083, 0.7267, and 0.7230. The policy may request a different K_t, so these are default-depth operating points. Default Top-3 has the highest QA despite lower retrieval F1 than the other conditions; Appendix C gives the process measurements.

5 Conclusion

REMIND treats access to a fixed personal memory store as a learned sequence of cue evocation, evidence-conditioned invocation, and stopping. It improves same-backbone Base RAG by 6.04 and 8.08 points at 4B and 8B, respectively; the 8B system also exceeds Search-R1. Type-level gains are uneven at 4B but cover all seven reported types at 8B. Retrieval F1 and QA peak at different operating points, while Top-3 is the best tested default depth. This supports optimizing the complete access-and-answer trajectory rather than retrieval coverage alone. Evaluation under matched retrieval budgets, repeated runs, and alternative memory organizations remains important future work.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jbNjgmE0OP.

Sam Audrain, Adrian W. Gilmore, Jenna M. Wilson, Daniel L. Schacter, and Alex Martin. A role for the anterior hippocampus in autobiographical memory construction regardless of temporal distance. *The Journal of Neuroscience*, 42(33):6445–6452, 2022. doi: 10.1523/JNEUROSCI.0832-22.2022. URL https://www.jneurosci.org/content/42/33/6445.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 3119–3137, 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Pratyay Banerjee, Masud Moshtaghi, Shivashankar Subramanian, Amita Misra, and Ankit Chadha. APEX-MEM: Agentic semi-structured memory with temporal reasoning for long-term conversational AI. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16470–16489. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.749. URL https://aclanthology.org/2026.acl-long.749/.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, et al. ReSearch: Learning to reason with search for LLMs via reinforcement learning. *arXiv preprint arXiv:2503.19470*, 2025. URL https://arxiv.org/abs/2503.19470.

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. URL https://arxiv.org/abs/2504.19413.

Sarah Daviddi, Tiziana Pedale, Peggy L. St. Jacques, Daniel L. Schacter, and Valerio Santangelo. Common and distinct correlates of construction and elaboration of episodic-autobiographical memory: An ALE meta-analysis. *Cortex*, 163:123–138, 2023. doi: 10.1016/j.cortex.2023.03.005. URL https://doi.org/10.1016/j.cortex.2023.03.005.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 1762–1777, 2023. doi: 10.18653/v1/2023.acl-long.99. URL https://aclanthology.org/2023.acl-long.99/.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515. PMLR, 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 3929–3938, 2020. URL https://proceedings.mlr.press/v119/guu20a.html.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *International Conference on Learning Representations*, 2026.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=haQD-7fF2Gc.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=jKN1pXi7b0.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *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, 2024.

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 *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.495. URL https://aclanthology.org/2023.emnlp-main.495/.

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. *arXiv preprint arXiv:2503.09516*, 2025.

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. 25961–25970, 2025. doi: 10.18653/v1/2025.emnlp-main.1318. URL https://aclanthology.org/2025.emnlp-main.1318/.

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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781, 2020. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

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 *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Junlin Liu, Jiangwang Chen, Zixin Song, Shuaiyu Zhou, Chunji Lv, Hank Wu, Kailin Jiang, Jinyang Wu, Bohan Yu, and Chenxi Zhou. From proprietary to open-source: Bridging the distribution gap via multi-agent protocol distillation in agentic search, 2026. URL https://arxiv.org/abs/2607.24280.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5303–5315. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.322. URL https://aclanthology.org/2023.emnlp-main.322/.

Yiwen Ma, Songjun Tu, Qichao Zhang, Dong Li, Linjing Li, and Dongbin Zhao. MemChain: Learning interpretable memory traces for memory-augmented LLM agents. *arXiv preprint arXiv:2607.24097*, 2026. URL https://arxiv.org/abs/2607.24097.

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. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. RaFe: Ranking feedback improves query rewriting for RAG. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 884–901, 2024. doi: 10.18653/v1/2024.findings-emnlp.49. URL https://aclanthology.org/2024.findings-emnlp.49/.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Rodrigo Nogueira and Kyunghyun Cho. Task-oriented query reformulation with reinforcement learning. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 574–583, 2017.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-Searcher: Incentivizing the search capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025. URL https://arxiv.org/abs/2503.05592.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 12991–13013, 2024. doi: 10.18653/v1/2024.acl-long.702. URL https://aclanthology.org/2024.acl-long.702/.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Rajan Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. 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. doi: 10.18653/v1/2025.acl-long.413. URL https://aclanthology.org/2025.acl-long.413/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.557. URL https://aclanthology.org/2023.acl-long.557/.

Liang Wang, Nan Yang, and Furu Wei. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9414–9423, 2023a. doi: 10.18653/v1/2023.emnlp-main.585. URL https://aclanthology.org/2023.emnlp-main.585/.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems*, 36:74530–74543, 2023b.

Yubo Wang, Qiuyu Zhao, Zenghui Sun, Shichao Dong, Jinsong Lan, Xiaoyong Zhu, Haoyang Li, Bo Zheng, and Lei Chen. Cmi-mem: Toward generalizable long-term memory management via cmi-augmented reinforcement learning. *arXiv preprint arXiv:2607.20553*, 2026.

Yujing Wang, Hainan Zhang, Liang Pang, Binghui Guo, Hongwei Zheng, and Zhiming Zheng. MaFeRw: Query rewriting with multi-aspect feedbacks for retrieval-augmented large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25434–25442, 2025. doi: 10.1609/aaai.v39i24.34732. URL https://ojs.aaai.org/index.php/AAAI/article/view/34732.

Zheng Wang, Zhongyang Li, Zeren Jiang, Dandan Tu, and Wei Shi. Crafting personalized agents through retrieval-augmented generation on editable memory graphs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4891–4906, 2024. doi: 10.18653/v1/2024.emnlp-main.281. URL https://aclanthology.org/2024.emnlp-main.281/.

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*, 2025.

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 *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 10000–10014. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.679. URL https://aclanthology.org/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 *Advances in Neural Information Processing Systems*, volume 38, pp. 17577–17604, 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024. URL https://arxiv.org/abs/2401.15884.

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 *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825. Association for Computational Linguistics, 2026. 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*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Xiaopeng Ye, Chen Xu, Chaoliang Zhang, Zhaocheng Du, Jun Xu, Gang Wang, and Zhenhua Dong. Q-PRM: Adaptive query rewriting for retrieval-augmented generation via step-level process supervision. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15113–15128, 2025. doi: 10.18653/v1/2025.findings-emnlp.817. URL https://aclanthology.org/2025.findings-emnlp.817/.

Jianing Yin and Tan Tang. DeferMem: Query-time evidence distillation via reinforcement learning for long-term memory QA. *arXiv preprint arXiv:2605.22411*, 2026. URL https://arxiv.org/abs/2605.22411.

Jing Yu, Yibo Zhao, Jiaming Zhang, and Xiang Li. LazyMem: Retrieve broadly, construct selectively for efficient long-term agent memory. *arXiv preprint arXiv:2607.22690*, 2026a. URL https://arxiv.org/abs/2607.22690.

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. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21457–21483. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.acl-long.981. URL https://aclanthology.org/2026.acl-long.981/.

Mengwei Yuan, Jianan Liu, Jing Yang, Xianyou Li, Weiran Yan, Yichao Wu, and Penghao Liang. TA-Mem: Tool-augmented autonomous memory retrieval for LLM in long-term conversational QA. *arXiv preprint arXiv:2603.09297*, 2026. URL https://arxiv.org/abs/2603.09297.

Qianhao Yuan, Jie Lou, Zichao Li, Jiawei Chen, Yaojie Lu, Hongyu Lin, Le Sun, Debing Zhang, and Xianpei Han. MemSearcher: Training LLMs to reason, search and manage memory via end-to-end reinforcement learning. *arXiv preprint arXiv:2511.02805*, 2025. URL https://arxiv.org/abs/2511.02805.

Kehao Zhang, Shangtong Gui, Sheng Yang, Wei Chen, and Yang Feng. Learning to remember: End-to-end training of memory agents for long-context reasoning. *arXiv preprint arXiv:2602.18493*, 2026. URL https://arxiv.org/abs/2602.18493.

AI Use Statement

Generative AI tools were used for code debugging, figure-caption and manuscript drafting, and language editing. The authors are responsible for the final content.

A REMIND Policy Execution

Algorithm 1 Evidence-conditioned memory invocation in REMIND.

Require: Question x, memory banks $\mathcal{M}$, policy π_θ, invocation budget $B = 5$
Ensure: Answer y or an invalid trajectory

$h \leftarrow (x, \text{memory summaries, optional temporal context})$
$n \leftarrow 0$
while true do
  Sample $a \sim \pi_\theta(\cdot \mid h)$
  if $a = \texttt{Memory_Finalize}(r)$ **then**
    Generate y from h, a, and the answer instruction
    return (y, τ)
  else if $a = \texttt{Memory_Invoke}(q, K)$ and $n < B$ **then**
    $o \leftarrow \mathcal{R}(q, \mathcal{M}, K)$
    Attach available source dialogue and remove duplicate chunks from o
    $h \leftarrow (h, a, o)$; $n \leftarrow n + 1$
  else
    return an invalid trajectory
  end if
end while

At training time, REMIND samples this complete trajectory and applies the reward in Equation 12. At inference time, the same loop runs without reward or judge calls.

B Notation

Notation	Meaning
x, y	Input question and final answer.
$\mathcal{M}^{c,e,s,p}$	Core, episodic, semantic, and procedural memory banks.
π_θ	REMIND policy parameterized by θ.
h_1, h_t	Initial context and policy history before action t.
a_t, o_t	Policy action and environment observation at turn t.
q_t, K_t	Search query and number of returned memories.
τ, T	Complete trajectory and its number of tool actions.
S_t, $\mathcal{G}$	Retrieved and gold evidence-session sets.
L, C_ℓ	Number of returned entries and candidate sessions for entry ℓ.
p_{ret}, r_{ret}	Per-entry precision and gold-session recall.
$\rho_{i,j}$, $\widehat{A}_i$	Token probability ratio and trajectory-level relative advantage.
R_{qa}, R_{ret}, R_{fmt}	Answer, retrieval, and valid-format rewards.
R_{evoke}, g_{new}, N_{dup}	Cue-grounding term, new-evidence gain, and redundant-query count.

Table 4: Notation used in the method. Superscripts c, e, s, p identify the four memory banks.

C Additional Configuration Measurements

Table 5 gives the query, retrieval, and answer measurements for the retrieval-depth comparison summarized in Section 4.3.

K	Edit dist.	ROUGE-L	Invokes	Ret. F1	QA
1	0.6284	0.2655	1.30	**0.6712**	0.7083
3	0.6917	0.2576	1.41	0.6307	**0.7267**
5	0.6306	0.2737	1.05	0.6635	0.7230

Table 5: Memory-access behavior across prompted default retrieval depths K, measured by query similarity, retrieval F1, and QA.

Default Top-1 uses 1.30 invocations on average, compared with 1.05 for default Top-5. The prompt specifies K, while each model-generated invocation can request a different K_t. The table also records edit distance and ROUGE-L, which describe how the generated queries differ from the original question.