ALL-IN-ONE MEMORY: PROCESS-REWARDED REINFORCEMENT LEARNING FOR MEMORY AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models increasingly rely on external memory to answer queries grounded in long-horizon histories such as multi-session dialogues and lengthy documents. However, existing memory methods often optimize retrieval and memory construction separately, without a shared reward space for evaluating search, writing, and stopping decisions. To address this, we propose **A**ll-**i**n-**O**ne **Mem**ory (**AiO-Mem**), which organizes these operations as transitions over a shared evidence state and directly evaluates the semantic quality of each action within a unified reward space. AiO-Mem compiles heterogeneous histories into source-traceable multi-view events, providing a shared substrate for retrieval and memory writing. A unified memory policy actor then rolls out complete interaction trajectories over this shared evidence state. A separate trajectory-conditioned process reward model is trained through evidence-grounded supervised fine-tuning and trajectory-level DPO to assign process scores to the collected trajectories. These scores are aggregated into an action-balanced trajectory reward and used by Process-Aligned DAPO (PA-DAPO) to optimize the memory policy. Across long-context memory benchmarks, AiO-Mem consistently improves answer accuracy over both static memory systems and RL-trained memory agents.

1 INTRODUCTION

Large language models (LLMs) have achieved remarkable progress across diverse natural language processing benchmarks (OpenAI, 2023; Touvron et al., 2023). Yet in long-context, multi-step reasoning scenarios, their static parametric knowledge and limited context windows remain critical bottlenecks (Liu et al., 2024). External memory-augmented architectures have therefore become important extensions of LLMs, ranging from dense retrieval systems (Lewis et al., 2020; Guu et al., 2020) and knowledge-graph memories (Edge et al., 2024) to persistent memory agents (Packer et al., 2023; Chhikara et al., 2025). Such systems rely on tightly coupled memory retrieval and memory construction: retrieval supplies relevant evidence, while construction converts it into compact working memory. Coordinating these stages is challenging because **the evolving memory state continually changes what evidence should be searched for, written, or discarded**.

Recent studies have introduced reinforcement learning (RL) into memory systems to enable self-improving memory behavior. As illustrated in Figure 1, existing RL-driven memory systems are typically optimized along two separate routes: retrieval-oriented methods, such as Search-R1 (Jin et al., 2025), learn how to search for evidence, while construction-oriented methods, such as Mem-α (Wang et al., 2025), learn when to insert, update, or merge memory entries. However, these methods still optimize retrieval and construction in isolation, leaving the two stages governed by separate objectives and evaluated through partially disconnected feedback signals. This separation exposes two key limitations. First, there is no shared evidence-state space in which retrieved evidence, written memories, and stopping decisions can be jointly evaluated, making it difficult to coordinate whether the system should continue searching, write a memory, skip redundant information, or stop. Second, existing supervision is largely outcome-centered: it assigns rewards to the final answer but provides little process-level credit for the intermediate memory operations that produced it. As a result, trajectories with the same final-answer score may still differ substantially in evidence relevance, memory faithfulness, redundancy control, and stopping sufficiency; **answer correctness alone cannot reliably evaluate the semantic quality of memory operations.**

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.

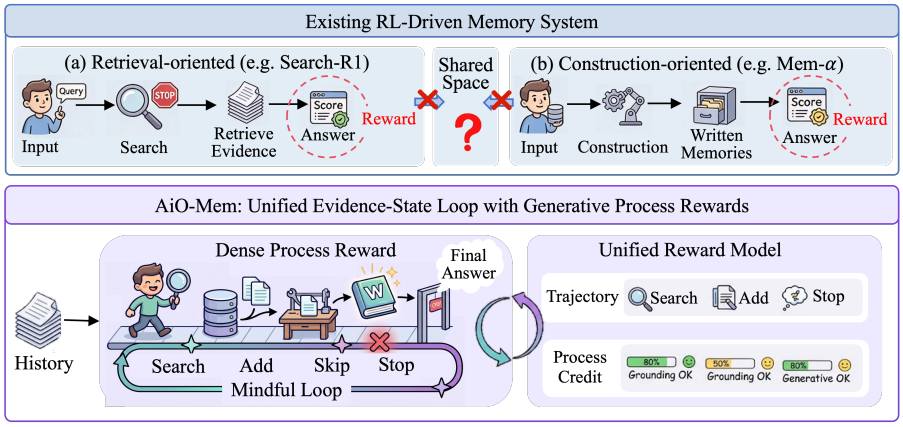


Figure 1: Comparison of existing RL-driven memory methods and AiO-Mem.

To overcome these limitations, we propose **A**ll-**i**n-**O**ne **Mem**ory (**AiO-Mem**), which unifies memory retrieval and construction as evidence-state transitions of a single memory-interaction agent, and uses a separate trajectory-conditioned reward model to evaluate the resulting trajectories. AiO-Mem first compiles heterogeneous histories into a traceable multi-view event bank, providing a shared source-grounded substrate for retrieval and memory writing. Built on this event bank, the memory policy actor operates over a shared evidence state and rolls out complete interaction trajectories through two sequential decision steps: *decide*, which determines whether to continue searching or stop, and *extract*, which determines whether retrieved candidates should be written into working memory or discarded. A unified process reward model is then trained through evidence-grounded SFT and trajectory-level DPO to assign process scores to each action in the collected trajectories, using rule-derived supervision and reasoning-model scores as complementary training signals. Finally, these process scores are aggregated into an action-balanced trajectory reward. Building on DAPO (Yu et al., 2026b), we propose Process-Aligned DAPO (PA-DAPO) to optimize the memory policy actor. We summarize our primary contributions as follows:

- We propose **AiO-Mem**[1], an all-in-one memory agent that unifies retrieval, writing, filtering, and termination within a single evidence-state transition loop. This formulation turns memory retrieval and construction from separately optimized modules into one jointly trainable policy over source-traceable evidence.
- We introduce a **trajectory-conditioned unified reward model** that directly assesses the semantic quality of each memory action within a shared evidence state, scoring search, add, skip, and stop in a common reward space. Action-balanced aggregation then converts these scores into a trajectory reward for memory-policy optimization with our proposed PA-DAPO.
- We conduct extensive experiments on long-context memory benchmarks, where AiO-Mem consistently improves both answer accuracy and reasoning faithfulness over static memory systems and RL-trained memory agents.

2 Related Work

Memory-Augmented Large Language Models. External memory equips LLMs with non-parametric knowledge beyond their context windows. Early retrieval-augmented models, such as RAG (Lewis et al., 2020) and REALM (Guu et al., 2020) established dense retrieval as a standard way to ground generation in external evidence. Later systems enrich retrieval with structured or task-specific memory organization: GraphRAG (Edge et al., 2024), HippoRAG (Gutiérrez et al., 2024), KG-RAG (Sanmartin, 2024), and Emotional RAG (Huang et al., 2024) incorporate entity graphs, graph traversal, knowledge-graph grounding, or affective similarity. A complementary line studies persistent memory across interactions. MemGPT (Packer et al., 2023) introduces virtual memory tiers; MemoryBank (Zhong et al., 2024) and Mem0 (Chhikara et al., 2025) maintain user-centric summaries; MemoryOS (Kang et al., 2025) and LightMem (Fang et al., 2026) organize memories

[1]The source code is available at https://anonymous.4open.science/r/Aio-Mem.

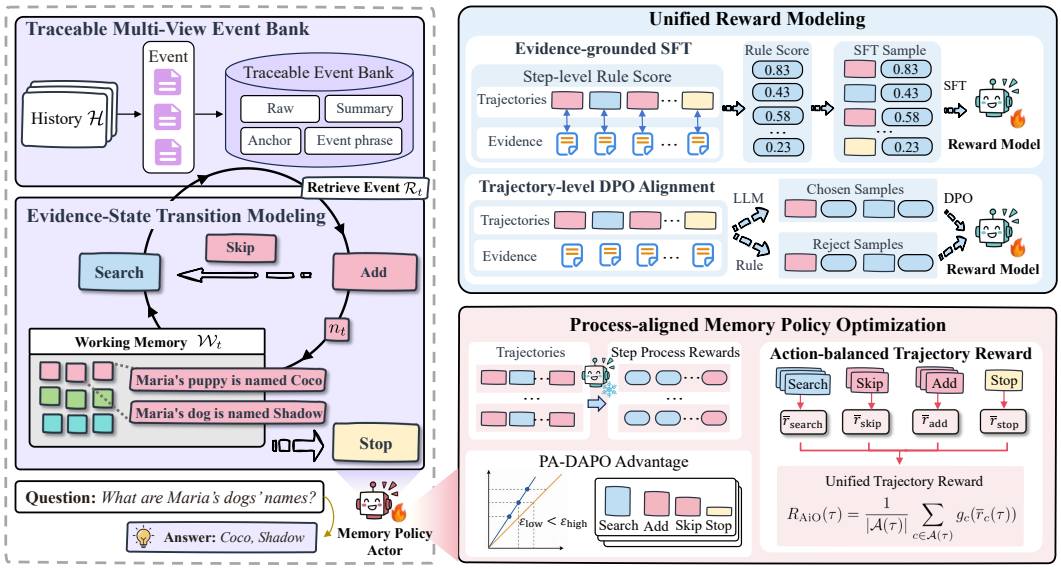


Figure 2: Overview of our AiO-Mem method.

into tiered or compressed stores; and A-Mem (Xu et al., 2026), Zep (Rasmussen et al., 2025), EverMemOS (Hu et al., 2026a), LangMem (LangChain, 2025), and GAM (Yan et al., 2025) explore atomic notes, MemCell structures, online consolidation, and iterative evidence gathering. These systems provide increasingly rich retrieval and construction mechanisms, but most coordinate memory operations through predefined workflows, hand-designed heuristics, or modular pipelines rather than a learned policy that jointly optimizes retrieval, writing, filtering, and termination.

Reinforcement Learning for Memory Policies. Recent work has applied RL to memory systems at different stages of the pipeline. Retrieval-oriented methods, including Search-R1 (Jin et al., 2025), R1-Searcher (Song et al., 2025), and DeepRetrieval (Jiang et al., 2025b), learn search or retrieval strategies that improve downstream answers. Persistent-memory methods, such as Memory-R1 (Yan et al., 2026), Mem-α (Wang et al., 2025), and AtomMem (Huo et al., 2026), optimize memory insertion, update, merging, or atomic memory management. More recent studies, including FineMem (Ma et al., 2026), TRUSTMEM (Yang et al., 2026), and Mem-T (Yue et al., 2026)), move toward process-level supervision for memory use. Despite this progress, most RL-based memory methods either rely primarily on final-answer correctness or provide process feedback within a restricted stage of the memory pipeline. As a result, they do not directly evaluate heterogeneous memory actions within a shared evidence state and a common reward space. AiO-Mem addresses this gap by learning a trajectory-conditioned reward model that assigns process-level semantic scores to different memory actions and uses them to optimize a unified memory policy.

3 Method

3.1 Problem Formulation and Method Overview

Problem Formulation. We study memory-augmented question answering (QA) over long-horizon histories. Let q be a query and $\mathcal{H} = (h_1, \ldots, h_N)$ be its associated history of N heterogeneous source units, such as dialogue sessions, document passages, or facts. Given $(q, \mathcal{H})$, a memory policy constructs a compact working memory $\mathcal{W}$ from which a frozen downstream generator $\hat{y} = \mathcal{G}(q, \mathcal{W})$. The learning objective is to acquire an optimal sequence of memory operations that transforms a dispersed history into a reasoning-sufficient evidence state.

Method Overview. As shown in Figure 2, we propose **AiO-Mem**, a four-component framework that trains a memory policy actor π_θ to construct working memory through sequential evidence-state transitions. AiO-Mem integrates four components for memory construction, interaction modeling, process supervision, and policy optimization: (i) *Traceable Multi-View Event Bank* (Section 3.2), which compiles $\mathcal{H}$ into source-traceable event units and provides the retrieval substrate for memory

interaction; (ii) *Evidence-State Transition Modeling* (Section 3.3), which formulates the interaction of π_θ as a finite-horizon MDP over a shared evidence-state space, with four action types organized into a *decide* step (**search** or **stop**) and, after each search, an *extract* step (**add** or **skip**); (iii) *Unified Reward Modeling* (Section 3.4), which trains a trajectory-conditioned reward model via evidence-grounded SFT and trajectory-level DPO to provide dense per-step process supervision; and (iv) *Process-Aligned Policy Optimization* (Section 3.5), which aggregates action-typed rewards into a balanced trajectory signal and optimizes π_θ through PA-DAPO.

3.2 Traceable Multi-View Event Bank

The source units in $\mathcal{H}$ often have incompatible granularities: multi-session dialogues are organized as turns, documents may spread a coherent events across multiple paragraphs, and database records are already atomic. Direct fixed-size chunking can therefore break event boundaries and detach memory entries from their source contexts. To preserve event integrity and source traceability, we apply a structure-aware compiler $\mathcal{C}$ to transform the history into a traceable event bank $\mathcal{B} = \mathcal{C}(\mathcal{H}) = \{e_m\}_{m=1}^{M}$. Each event is defined as $e_m = (\{x_m^v\}_{v\in\mathcal{V}}, \ell_m)$, where $\mathcal{V} = \{\mathrm{raw}, \mathrm{sum}, \mathrm{anc}, \mathrm{phr}\}$ denotes four complementary retrieval views and $\ell_m = (n, i_{\mathrm{start}}, i_{\mathrm{end}})$ is a provenance locator that anchors the event to its source unit h_n. Specifically, The compiler first partitions $\mathcal{H}$ along topic-shift boundaries, then constructs four views for each event. The raw view x_m^{raw} preserves lexical details, summary x_m^{sum} captures macroscopic semantic coherence, query-anchor x_m^{anc} encodes hypothetical queries the event can answer, and event-phrase x_m^{phr} provides a dense, low-noise semantic label.

Since no single representation captures lexical, semantic, and query-intent cues simultaneously, for each search query u_t issued by the π_θ, we fuse view-specific rankings via equal-weight Reciprocal Rank Fusion (Cormack et al., 2009):

$$\mathrm{score}(e_m \mid u_t) = \sum_{v\in\mathcal{V}} \frac{1}{\kappa + \mathrm{rank}_v(e_m \mid u_t)},$$

where $\kappa > 0$ is a smoothing constant. Dense retrieval is used for x^{sum} and x^{phr}, BM25 for x^{anc}, and a hybrid of both for x^{raw}. The retrieval operator returns the top-K_{ret} events as candidate list $\mathcal{R}(u_t)$, with each retaining its locator ℓ_m for source-verifiable memory writing and reward evaluation.

3.3 Evidence-State Transition Modeling

A single retrieval pass is often insufficient when the answer depends on evidence scattered across the history. We therefore formulate memory interaction as a finite-horizon Markov Decision Process (MDP) over evidence states. The memory policy actor π_θ sequentially selects memory operations and the process is specified by a state space $\mathcal{S}$, an action space $\mathcal{A}$, a deterministic transition function $\mathcal{F}$, and a maximum horizon $T_{\max}$.

At step $t \in \{1, \ldots, T\}$, the evidence state is $z_t = (q, \mathcal{W}_t, \mathcal{R}_t) \in \mathcal{S}$, where q is the query, $\mathcal{W}_t$ is the current working memory, $\mathcal{R}_t$ is the latest candidate list, and $m_t \in \{\mathrm{decide}, \mathrm{extract}\}$ indicates the current decision mode. The memory policy actor samples an action $a_t \sim \pi_\theta(\cdot \mid z_t)$ under a mode-specific action mask, and $c_t = \mathrm{type}(a_t)$ denotes its action type. In the *decide* mode, the policy chooses either **search**(u_t), which issues a search query u_t to the event bank and retrieves $\mathcal{R}(u_t)$, or **stop**, which terminates the episode. After a search, the process enters the *extract* mode, where the policy chooses either **add**(n_t), which appends a source-bound note n_t to the working memory, or **skip**, which discards the current candidates. This constrained structure prevents invalid operations, such as adding a note before retrieval or issuing a new search before deciding whether to use the current candidates. The four action types $\{\mathrm{search}, \mathrm{add}, \mathrm{skip}, \mathrm{stop}\}$ induce the following deterministic transitions:

$$z_{t+1} = \mathcal{F}(z_t, a_t; \mathcal{B}) = \begin{cases} (q, \mathcal{W}_t, \mathcal{R}(u_t)), & c_t = \mathrm{search} \\ (q, \mathcal{W}_t \oplus [n_t], \mathcal{R}_t), & c_t = \mathrm{add} \\ (q, \mathcal{W}_t, ()), & c_t = \mathrm{skip} \\ (q, \mathcal{W}_t, \mathcal{R}_t), & c_t = \mathrm{stop} \end{cases} \tag{1}$$

The episode ends when the memory policy actor selects **stop**, or when the maximum horizon $T_{\max}$ is reached. The final working memory is passed to the frozen generator to produce $\hat{y} = \mathcal{G}(q, \mathcal{W}_T)$.

Each interaction trajectory is written as $\tau = (z_1, a_1, \ldots, z_T, a_T, z_{T+1})$, and the collected trajectories form $\mathcal{D}_{\text{traj}}$, which is used for subsequent reward modeling and policy optimization.

3.4 Unified Reward Modeling

Final-answer correctness provides only terminal feedback and can not identify which memory operations make a trajectory useful. We therefore train a generative reward model $\mathcal{J}_\phi$ to assign dense per-step process rewards in complete memory trajectories. Unlike the memory policy actor, $\mathcal{J}_\phi$ evaluates the resulting actions and provides process supervision without affecting state transitions.

Stage 1: Evidence-grounded SFT. Let $\mathcal{E}^\star$ denote the gold evidence spans and $\mathrm{Src}(\cdot)$ the provenance projection that maps an event or note to its referenced raw source spans. At step t, the uncovered gold evidence is $\mathcal{E}_t^{\text{miss}} = \mathcal{E}^\star \setminus \mathrm{Src}(\mathcal{W}_t)$. For a search action, the newly exposed gold evidence is $\Delta_t^{\text{ret}} = \mathcal{E}_t^{\text{miss}} \cap \mathrm{Src}(\mathcal{R}(u_t))$, and for add action, the newly commits gold evidence is $\Delta_t^{\text{write}} = \mathcal{E}_t^{\text{miss}} \cap \mathrm{Src}(n_t)$. We use action-specific deterministic rules over these evidence-grounded information to compute a step-level rule score r_t^{rule} for each action type $c \in \mathcal{A}$. For each trajectory τ produced by the memory policy actor, we serialize the trajectory as $x_\tau = \mathrm{Serialize}(y^\star, \mathcal{E}^\star, \tau)$, and fine-tune $\mathcal{J}_\phi$ on the rule reward sequence $\mathbf{v}_b^{\text{rule}} = \{(c_t, r_t^{\text{rule}})\}$:

$$\mathcal{L}_{\text{SFT}}(\phi) = -\frac{1}{\sum_b L_b^{\text{RM}}} \sum_b \sum_t \log p_\phi\big(v_{b,t}^{\text{rule}} \mid x_b, \mathbf{v}_{b,<t}^{\text{rule}}\big) \tag{2}$$

where L_b^{RM} the target token length, and p_ϕ the autoregressive output distribution of $\mathcal{J}_\phi$.

Stage 2: Trajectory-level DPO alignment. Rule-derived scores provide action-level supervision based on evidence coverage and provenance, but do not directly assess the semantic quality of an action in its current evidence state. We therefore provide a reasoning-capable model with the full trajectory context x_τ and have it independently score each step via chain-of-thought reasoning, producing $\mathbf{v}_b^{\text{CoT}}$. We treat $\mathbf{v}_b^+ = \mathbf{v}_b^{\text{CoT}}$ as chosen and $\mathbf{v}_b^- = \mathbf{v}_b^{\text{rule}}$ as rejected, forming preference dataset $\mathcal{D}_{\text{pref}} = \{(x_b, \mathbf{v}_b^+, \mathbf{v}_b^-)\}$, and minimize the DPO loss with frozen reference ϕ_{ref} and strength $\beta > 0$:

$$\mathcal{L}_{\text{DPO}}(\phi) = -\mathbb{E}\left[\log \mathrm{sigmoid}\left(\beta \log \frac{p_\phi(\mathbf{v}^+ \mid x)}{p_{\phi_{\text{ref}}}(\mathbf{v}^+ \mid x)} - \beta \log \frac{p_\phi(\mathbf{v}^- \mid x)}{p_{\phi_{\text{ref}}}(\mathbf{v}^- \mid x)}\right)\right] \tag{3}$$

After training, $\mathcal{J}_\phi$ directly outputs $\{(c_t, \hat{r}_t)\}_{t=1}^T$, where $\hat{r}_t \in \mathbb{R}$ assesses the semantic quality of action c_t in its current evidence state.

3.5 Process-Aligned Memory Policy Optimization

AiO-Mem unifies heterogeneous memory action types within a shared evidence-state transition process. We aggregate their step-level rewards into an action-balanced trajectory signal, averaging within action types before integrating across types. For each category $c \in \mathcal{A}$, let $\mathcal{I}_c(\tau) = \{t : c_t = c\}$ be the steps where c occurs, $\overline{r}_c(\tau) = |\mathcal{I}_c(\tau)|^{-1} \sum_{t \in \mathcal{I}_c(\tau)} \hat{r}_t$ the within-category mean, and $g_c(r) = (r - r_c^{\text{lo}})/(r_c^{\text{hi}} - r_c^{\text{lo}})$ a normalization to $[0, 1]$ using the rubric bounds r_c^{lo} and r_c^{hi}. Denoting by $\mathcal{A}(\tau) = \{c \in \mathcal{A} : \mathcal{I}_c(\tau) \neq \varnothing\}$ the categories present in τ, the unified trajectory reward is:

$$R_{\text{AiO}}(\tau) = \frac{1}{|\mathcal{A}(\tau)|} \sum_{c \in \mathcal{A}(\tau)} g_c(\overline{r}_c(\tau)) \tag{4}$$

PA-DAPO. We optimize the memory policy actor π_θ with our proposed PA-DAPO. Building on DAPO (Yu et al., 2026b), AiO-Mem extends the loss computation from the conventional $(\texttt{group}, \texttt{token})$ structure to a new $(\texttt{group}, \texttt{action}, \texttt{token})$ hierarchy over trajectories. Formally, let $\mathcal{D}_{\text{RL}}$ denote the distribution of query–history pairs used for policy optimization. For $(q, \mathcal{H}) \sim \mathcal{D}_{\text{RL}}$, the old memory policy actor samples G trajectories $\{\tau_j\}$ with rewards $R_j = R_{\text{AiO}}(\tau_j)$. The group-relative advantage $\hat{A}_j = (R_j - \mu)/\sigma$ is computed from the within-group mean μ and standard deviation σ. Let $o_{j,k,t}$ be the t-th policy-generated token of the k-th action step in τ_j, $\mathbf{h}_{j,k,t}$ the preceding context, and $L_{j,k}$ the step's token count. PA-DAPO maximizes:

$$\mathcal{J}_{\text{PA-DAPO}}(\theta) = \mathbb{E}\left[\frac{\sum_j \sum_k \sum_t \Gamma_{j,k,t}(\theta)}{\sum_j \sum_k L_{j,k}}\right], \quad \sigma > 0 \tag{5}$$

Table 1: Main results on **LoCoMo**.

Method	Multi-hop		Temporal		Open-domain		Single-hop		Overall	
	F1	B1	F1	B1	F1	B1	F1	B1	F1	B1
Reference										
Long-Context	23.2	16.8	19.0	13.4	13.9	10.7	40.7	31.5	31.5	23.9
RAG	23.5	17.1	43.1	37.4	20.2	14.9	49.5	44.9	41.6	36.5
Agent Memory Harness										
MemGPT	16.7	14.0	12.6	10.9	11.6	9.2	14.0	11.8	14.1	11.8
MemoryBank	25.5	19.4	9.2	7.4	16.4	12.4	26.7	17.7	22.3	15.7
Mem0	35.4	27.4	46.8	39.5	26.6	21.0	47.3	40.7	43.7	36.8
MemoryOS	35.2	27.3	41.0	32.7	22.1	17.9	48.4	42.6	42.8	36.3
LightMem	30.8	25.8	44.7	40.7	18.9	14.4	43.8	38.8	40.0	35.3
A-Mem	27.2	21.1	43.9	36.0	15.4	12.7	44.6	38.3	39.4	33.0
GAM	32.2	28.7	26.3	22.5	18.5	14.5	32.2	25.5	30.2	24.8
Trained Method										
MEM1	19.0	15.6	30.5	23.5	14.2	11.4	27.5	22.1	25.7	20.5
MemAgent	27.9	22.7	37.9	31.9	20.3	16.5	35.9	29.6	33.8	28.0
Mem-T	45.1	36.6	65.1	**60.1**	33.0	28.9	63.8	58.0	58.7	52.6
Ours										
AiO-Mem	**47.1**	**38.8**	**66.0**	59.4	**33.7**	**30.5**	**67.0**	**61.5**	**61.0**	**55.0**

where $\Gamma_{j,k,t}(\theta) = \min(\rho_{j,k,t}\hat{A}_j, \text{clip}(\rho_{j,k,t}, 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}})\hat{A}_j)$ is the clipped surrogate, $\rho_{j,k,t}(\theta) = \pi_\theta(o_{j,k,t} \mid \mathbf{h}_{j,k,t})/\pi_{\theta_{\text{old}}}(o_{j,k,t} \mid \mathbf{h}_{j,k,t})$ the token-level importance ratio, and $\epsilon_{\text{low}}, \epsilon_{\text{high}}$ the clipping bounds. The constraint $\sigma > 0$ retains only queries with reward variation, filtering fully converged samples. Operating on an action-balanced trajectory reward rather than binary answer correctness, PA-DAPO provides a graded training signal for end-to-end policy optimization.

4 Experiment

4.1 Experimental Setting

Datasets and Evaluation We evaluate AiO-Mem on three complementary memory benchmarks, including LoCoMo (Maharana et al., 2024), MemoryAgentBench (Hu et al., 2026b), and PersonaMem (Jiang et al., 2025a). LoCoMo evaluates long-range conversational memory across multi-session interactions. Following MEM-T (Yue et al., 2026), we adopt the same data configuration by splitting the LoCoMo dataset into training, validation, and test sets with a ratio of 1:1:8 to ensure a fair comparison. MemoryAgentBench valuates memory agents under incremental multi-turn interactions across diverse tasks. Following Fine-Mem (Ma et al., 2026), we adopt the same evaluation scope and baselines to assess out-of-distribution generalization. PersonaMem further extends evaluation from factual memory to dynamic user profiling. More details are shown in Appendix C.1.

Baselines. Twenty systems in three groups. ① *Reference:* Long-Context and RAG. ② *Agent Memory Harness:* MemGPT (Packer et al., 2023), MemoryBank (Zhong et al., 2024), Mem0 (Chhikara et al., 2025), MemoryOS (Kang et al., 2025), LightMem (Fang et al., 2026), A-Mem (Xu et al., 2026), GAM (Yan et al., 2025), Zep (Rasmussen et al., 2025), LangMem (LangChain, 2025), EverMemOS (Hu et al., 2026a), and Emotional RAG (Huang et al., 2024). ③ *RL-Trained Method:* MEM1 (Zhou et al., 2026), MemAgent (Yu et al., 2026a), Mem-T (Yue et al., 2026), AtomMem (Huo et al., 2026), Mem-α (Wang et al., 2025), FineMem (Ma et al., 2026), and TrustMem (Yang et al., 2026). Details are provided in Appendix C.1.

Experimental Details. We instantiate AiO-Mem with Qwen3-8B as the reward model and Qwen3-4B as the memory policy actor. BGE-M3[2] and BM25 are used for multi-view event retrieval. For policy training, we set $K_{\text{ret}} = 10$, $T_{\max} = 8$. For reward-model SFT, we train for 2 epochs with a batch size of 8 and a learning rate of 1×10^{-4}. Trajectory-level DPO is then performed for 1 epoch with a batch size of 8, a learning rate of 2×10^{-6}, and $\beta = 0.1$. For PA-DAPO, we sample 8 trajectories per query and optimize the memory policy with a learning rate of 2×10^{-5}. More training setup and parameter configurations are listed in Appendix C.2.

[2]https://huggingface.co/BAAI/bge-m3.

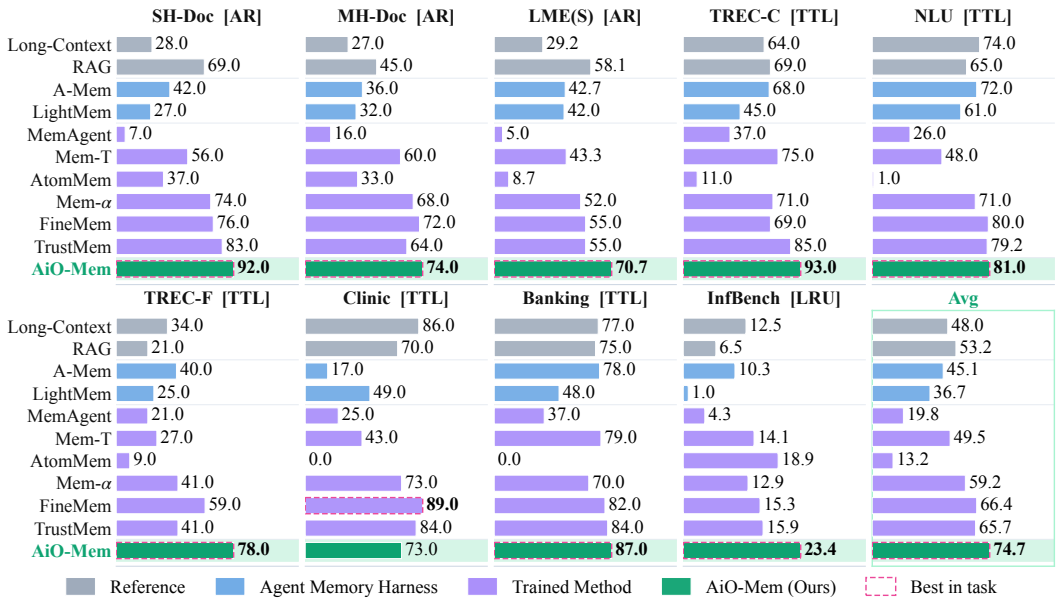


Figure 3: Main results on **MemoryAgentBench**.

4.2 Main Results

LoCoMo. As shown in Table 1, , AiO-Mem achieves the strongest overall performance and consistently outperforms competing methods across diverse question types. The gains span both evidence-intensive reasoning and direct memory queries, reflecting a coherent improvement pattern rather than isolated advantages on specific categories. In contrast, existing memory systems exhibit more uneven performance across different reasoning demands. These results suggest that **jointly modeling heterogeneous memory actions under shared evidence states and unified process supervision enables more reliable memory interaction throughout the reasoning process**.

MemoryAgentBench. As shown in Figure 3, AiO-Mem achieves the strongest overall performance across MemoryAgentBench, with gains that form a consistent pattern across retrieval, test-time learning, and long-range understanding. It achieves an average score of 74.7 across the nine tasks, outperforming Fine-Mem, the strongest baseline, by 8.3 points. Compared with the best competing result on each task, AiO-Mem improves LME(S), TREC-F, and InfBench by 12.6, 19.0, and 4.5 points, respectively. Such gains across heterogeneous task formats demonstrate **strong out-of-distribution generalization beyond the LoCoMo training setting**. This further supports the design of AiO-Mem for generalizable memory interaction.

PersonaMem. AiO-Mem achieves the best overall performance on PersonaMem, surpassing the strongest baseline by 3.9 points and **demonstrating stronger long-term user profiling and personalization**. More details are shown in Appendix C.1.

4.3 Ablation Study

To examine the contribution of each component in AiO-Mem, we conduct ablation experiments by removing individual modules, as shown in Table 2. Removing Event Compilation (*w/o Event Comp.*) or Multi-view Retrieval (*w/o Multi-view Ret.*) consistently degrades performance, showing that structured event organization and complementary retrieval cues provide a reliable evidence basis for memory interaction. Removing Working-Memory Construction (*w/o WM Constr.*) causes the most substantial degradation, highlighting its role in progressively transforming retrieved evidence into reasoning-sufficient working memory. For Unified Reward Modeling, removing Evidence-Grounded SFT (*w/o Evid. SFT*) weakens performance, demonstrating the importance of evidence-based supervision for establishing reliable process rewards. Further removing Trajectory-Level DPO (*w/o Traj. DPO*) also leads to consistent degradation, showing that trajectory-level alignment im-

Table 2: Ablation on **LoCoMo**. Row abbreviations: Event Comp. = Event Compilation, Multi-view Ret. = Multi-view Retrieval, WM Constr. = Working-Memory Construction, Evid. SFT = Evidence-Grounded SFT, and Traj. DPO = Trajectory-Level DPO. Red values denote drops over AiO-Mem.

Method	Multi-hop		Temporal		Open-domain		Single-hop		Overall	
	F1	B1	F1	B1	F1	B1	F1	B1	F1	B1
Base Model	39.3	30.9	65.0	58.4	25.9	21.9	61.3	56.1	55.8 (-5.2)	49.8 (-5.2)
Memory Representation and Interaction Ablations										
w/o Event Comp.	43.2	34.6	66.9	61.0	27.2	22.6	64.3	58.7	58.6 (-2.4)	52.5 (-2.5)
w/o Multi-view Ret.	41.3	33.4	65.8	60.4	29.5	24.4	64.2	58.6	58.1 (-2.9)	52.1 (-2.9)
w/o WM Constr.	33.2	26.4	57.0	50.9	24.3	20.0	57.6	52.3	50.9 (-10.1)	45.2 (-9.8)
Unified Reward Modeling Ablations										
w/o Evid. SFT	41.4	33.6	68.4	60.6	28.6	24.5	62.9	57.4	57.9 (-3.1)	51.6 (-3.4)
w/o Traj. DPO	43.4	35.2	66.5	59.5	34.2	29.9	64.0	58.4	58.8 (-2.2)	52.6 (-2.4)
AiO-Mem	**47.1**	**38.8**	**66.0**	**59.4**	**33.7**	**30.5**	**67.0**	**61.5**	**61.0**	**55.0**

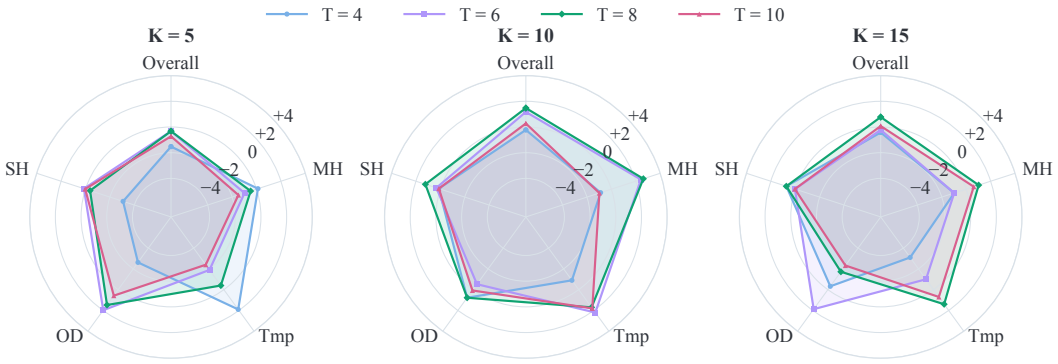


Figure 4: Parameter analysis on **LoCoMo** over top-K ($K \in \{5, 10, 15\}$) and maximum interaction steps $T_{\max} \in \{4, 6, 8, 10\}$, reporting F1 across four question types (MH = Multi-hop, Tmp = Temporal, OD = Open-domain, SH = Single-hop) and overall. To facilitate comparison, all values are normalized for visualization on the chart.

proves reward estimation beyond rule-based supervision. Overall, AiO-Mem achieves the best performance by jointly integrating these components within a unified memory-interaction framework.

4.4 Analysis of Complexity and Sensitivity

Complexity Analysis. Existing memory systems repeatedly summarize and rewrite accumulated memories, incurring a history-dependent cost of $O(N\ell + NM_1R_1)\mathcal{C}_{\text{LLM}}$. In contrast, AiO-Mem decouples persistent storage from query-time working-memory construction. The event bank $\mathcal{B}$ is built in a single pass, while query-time reasoning is bounded by K_{ret} and $T_{\max}$, yielding $O(N\ell + T_{\max}K_{\text{ret}}\ell_e)\mathcal{C}_{\text{LLM}}$. By bounding the working context to a constant size, AiO-Mem transforms a history-dependent scaling cost into a local, horizon-bounded overhead that remains invariant to the stream length. Empirically, this decoupling mechanism reduces retrieval and construction token consumption by 79.9% and 73.8% respectively, bypassing the active memory maintenance bottleneck of traditional architectures. Detailed derivations are provided in Appendix F.

Sensitivity Analysis. We further examine the robustness of AiO-Mem to two key hyperparameters, the retrieval width K_{ret} and maximum interaction horizon $T_{\max}$. As shown in Figure 4, performance remains relatively stable across configurations, while increasing either parameter yields no consistent gains. A moderate retrieval width provides sufficient evidence coverage without introducing excessive irrelevant candidates, while an appropriate interaction horizon enables effective multi-step memory interaction without redundant actions. The configuration $(K_{\text{ret}}, T_{\max}) = (10, 8)$ achieves the strongest overall performance, suggesting that AiO-Mem benefits from a balanced retrieval and interaction budget. Detailed F1 and B1 results are provided in Appendix D.1.

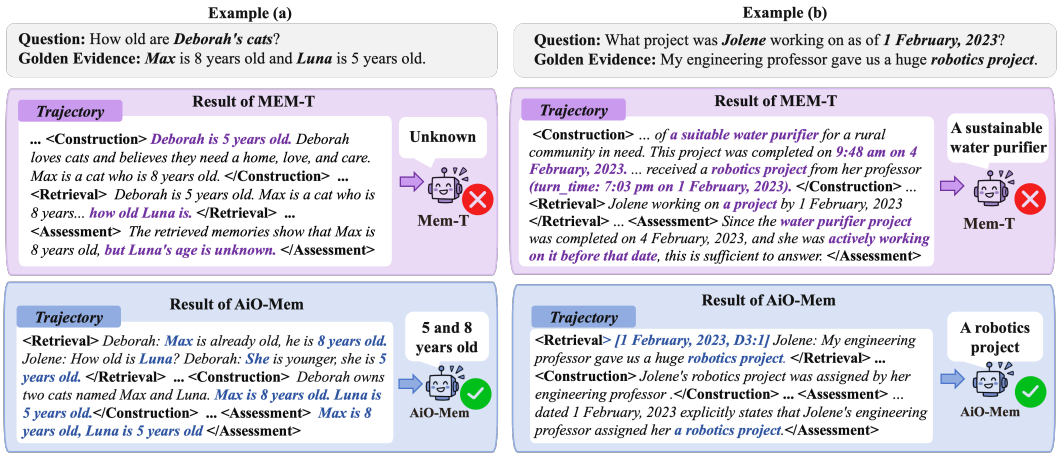


Figure 5: Two cases between the MEM-T and our AiO-Mem method.

4.5 Case Study

To further illustrate the advantage of AiO-Mem over Mem-T, Figure 5 presents two memory-interaction cases. In Example (a), Mem-T incorrectly binds "she is 5 years old" to Deborah rather than Luna during memory construction. Although relevant evidence is retrieved, the incorrect binding is retained in memory and leads to an incorrect assessment. In contrast, AiO-Mem retrieves source-grounded evidence and constructs a working-memory note that preserves the correct entity relation. In Example (b), Mem-T conflates the temporally distinct robotics and water-purifier projects, causing the assessment to overlook the query-time cutoff and select evidence from a later event. AiO-Mem instead retrieves dated evidence and preserves its temporal relation during working-memory construction, leading to the correct answer. These cases show that AiO-Mem improves memory interaction by jointly coordinating retrieval and construction under shared evidence states and process-level supervision, preventing errors from propagating into later reasoning.

5 Additional Ablations in Appendix

We further examine AiO-Mem across the following key dimensions. **(i) Multi-View Retrieval Complementarity and Sensitivity.** The four retrieval views provide complementary evidence coverage, jointly covering 97.03% of question–evidence pairs while remaining robust to retrieval configurations and context cost (**Appendix D.3**, **Appendix D.4**).**(ii) Unified Reward Alignment.** The trained reward model shows stronger correlations with evidence Recall and Precision, better reflecting evidence-grounded trajectory quality (**Appendix E.1**). **(iii) Trajectory-level DPO Alignment Study.** We visualize how trajectory-level DPO distinguishes trajectories with identical Evidence-ID metrics through finer-grained semantic judgments(**Appendix E.2**). **(iv) Training Dynamics.** Overall, consistent gains across Search, Extract, and Answer-Decision rewards indicate stable PA-DAPO training and coordinated step-wise optimization(**Appendix E.3**). **(v) Process-Aligned Evidence Flow.** PA-DAPO improves evidence retrieval, working-memory quality, and Answer F1 across the full pipeline (**Appendix E.4**).

6 Conclusion

Existing memory systems often optimize retrieval and memory construction separately, making it difficult to coordinate heterogeneous memory operations under a shared objective. To address this, we introduce AiO-Mem, which unifies retrieval, writing, filtering, and termination as transitions over a shared evidence state, and learns a trajectory-conditioned reward model to provide process-level supervision for memory-policy optimization. Experiments across long-context memory benchmarks show that AiO-Mem consistently outperforms both static memory systems and RL-trained memory agents, with strong out-of-distribution generalization beyond conversational QA. These results highlight the promise of an All-in-One formulation for learning reliable and generalizable memory interaction in LLM agents.

AI USE STATEMENT

LLMs were used to generate data and to evaluate memory-interaction trajectories as judges, as described in the paper. The core method development in this paper does not involve LLM usage. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

AiO-Mem organizes long interaction histories into retrievable events and constructs query-specific working memory. Although this can improve an agent's ability to use historical evidence, such histories may contain sensitive information. Retaining source-linked events may increase the risk of inappropriate access to or disclosure of that information. Inaccurate or biased information in a history may also be retrieved and propagated into later answers. Our experiments use the existing LoCoMo, MemoryAgentBench, and PersonaMem benchmarks. Deployment with real user histories would require appropriate safeguards for user consent, data access, retention and deletion, as well as mechanisms for identifying and correcting inaccurate memories.

REPRODUCIBILITY STATEMENT

The method is described in Section 3. Appendix C provides the experimental details. Prompts and memory-agent tool calls are provided in Appendix G. The source code is available in an anonymous repository at this anonymous repository.

REFERENCES

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.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 758–759, 2009.

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

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, et al. Lightmem: Lightweight and efficient memory-augmented generation. In *International Conference on Learning Representations*, volume 2026, pp. 98706–98729, 2026.

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, et al. Evermemos: A self-organizing memory operating system for structured long-horizon reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45836–45853, 2026a.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in llm agents via incremental multi-turn interactions. In *International Conference on Learning Representations*, volume 2026, pp. 156259–156291, 2026b.

Le Huang, Hengzhi Lan, Zijun Sun, Chuan Shi, and Ting Bai. Emotional rag: Enhancing role-playing agents through emotional retrieval. In *2024 IEEE International Conference on Knowledge Graph (ICKG)*, pp. 120–127. IEEE, 2024.

Yupeng Huo, Yaxi Lu, Zhong Zhang, Haotian Chen, and Yankai Lin. Atommem : Learnable dynamic agentic memory with atomic memory operation, 2026. URL https://arxiv.org/abs/2601.08323.

Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J Taylor, and Dan Roth. Know me, respond to me: Benchmarking llms for dynamic user profiling and personalized responses at scale. *arXiv preprint arXiv:2504.14225*, 2025a.

Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, and Jiawei Han. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. *arXiv preprint arXiv:2503.00223*, 2025b.

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.

Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

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

LangChain. LangMem: Long-term memory for language agents. https://github.com/langchain-ai/langmem, 2025. Official software repository and documentation, accessed 2026-09-19.

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

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

Weitao Ma, Xiaocheng Feng, Lei Huang, Xiachong Feng, Zhanyu Ma, Jun Xu, Jiuchong Gao, Jinghua Hao, Renqing He, and Bing Qin. Fine-mem: Fine-grained feedback alignment for long-horizon memory management. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19666–19684, 2026.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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.

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

Diego Sanmartin. Kg-rag: Bridging the gap between knowledge and creativity. *arXiv preprint arXiv:2405.12035*, 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.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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

Joe H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026.

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

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, et al. 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, 2026.

Tianyu Yang, Sudipta Paul, Vijay Srinivasan, Vivek Kulkarni, and Srinivas Chappidi. Trustmem: Learning trustworthy memory consolidation for llm agents with long-term memory. *arXiv preprint arXiv:2606.25161*, 2026.

Xiaoju Ye. calflops: a flops and params calculate tool for neural networks in pytorch framework. `https://github.com/MrYxJ/calculate-flops.pytorch`, 2023.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. In *International Conference on Learning Representations*, volume 2026, pp. 39458–39486, 2026a.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026b.

Yanwei Yue, Boci Peng, Xuanbo Fan, Jiaxin Guo, Qiankun Li, and Yan Zhang. Mem-t: Densifying rewards for long-horizon memory agents. *arXiv preprint arXiv:2601.23014*, 2026.

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

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Bryan Kian Hsiang Low, and Paul Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. In *International Conference on Learning Representations*, volume 2026, pp. 58413–58438, 2026.

A Appendix Overview and Organization

This appendix provides supplementary details to support and extend the main paper. The organization of the appendix is as follows:

1. **Additional Related Work (Section B):** This section provides further discussion of memory-augmented large language models and reinforcement learning methods for memory policies, with particular emphasis on persistent memory management and process-level memory optimization.
2. **Experimental Setup and Supplementary Experiments (Sections C and D):** These sections describe the datasets, baselines, and implementation details, and provide additional experiments on parameter sensitivity, PersonaMem, multi-view retrieval complementarity, and inference-time retrieval configurations.
3. **Reward, Optimization, and Efficiency Analysis (Sections E and F):** These sections further analyze reward alignment, qualitative reward cases, PA-DAPO training dynamics, evidence flow through the memory pipeline, as well as the computational complexity and inference efficiency of AiO-Mem.
4. **Prompts and Tool Calls (Section G):** This section provides the memory-agent tool definitions and prompt templates used for memory interaction and stepwise reward modeling, supporting reproducibility of the proposed framework.

B Related Work

B.1 Additional Discussion on Memory-Augmented Large Language Models

Memory-augmented language models have evolved from external retrieval toward persistent and structured memory management. Early retrieval-augmented methods, including RAG (Lewis et al., 2020) and REALM (Guu et al., 2020), introduce non -parametric knowledge through dense retrieval, while subsequent systems such as GraphRAG (Edge et al., 2024), HippoRAG (Gutiérrez et al., 2024), KG-RAG (Sanmartin, 2024), and Emotional RAG (Huang et al., 2024) enrich retrieval with graph structures, multi-step exploration, or task-specific similarity signals. These methods primarily focus on improving how external evidence is accessed for a given query. Another line of work develops persistent memory across interactions. MemGPT (Packer et al., 2023), MemoryBank (Zhong et al., 2024), and Mem0 (Chhikara et al., 2025) introduce explicit mechanisms for storing, updating, and maintaining long-term information, while MemoryOS (Kang et al., 2025) and LightMem (Fang et al., 2026) improve scalability through hierarchical storage and lightweight consolidation. More recent systems, including A-Mem (Xu et al., 2026), Zep (Rasmussen et al., 2025), EverMemOS (Hu et al., 2026a), LangMem (LangChain, 2025), and GAM (Yan et al., 2025), further explore atomic memory units, temporal structures, background consolidation, and iterative evidence gathering. These methods substantially expand the capabilities of long-term memory systems in construction, organization, updating, and retrieval. However, the individual operations are still largely coordinated through predefined workflows, prompts, or heuristic rules, rather than being **jointly optimized according to their contribution to downstream evidence acquisition**.

B.2 Additional Discussion on Reinforcement Learning for Memory Policies

Reinforcement learning has increasingly been introduced to replace manually specified memory behaviors with learned policies. Early efforts mainly focus on retrieval, where Search-R1 (Jin et al., 2025), R1-Searcher (Song et al., 2025), and DeepRetrieval (Jiang et al., 2025b) optimize multi-step search, tool invocation, or query generation through task- or retrieval-oriented rewards. Another line applies RL to recurrent memory states: MEM1 (Zhou et al., 2026) and MemAgent (Yu et al., 2026a) learn to continually compress or overwrite bounded memory representations over long contexts and propagate downstream rewards to preceding memory decisions. Persistent-memory methods such as Memory-R1 (Yan et al., 2026), Mem-α (Wang et al., 2025), and AtomMem (Huo et al., 2026) further extend the learned policy from memory construction and updating toward broader combinations of reading, editing, and answering. More recent work introduces finer-grained supervision to improve credit assignment across long memory-operation sequences. Fine-Mem (Ma et al., 2026)

and TRUSTMEM (Yang et al., 2026) provide localized feedback for memory construction or updating through intermediate QA signals, downstream usage, or verifier-based evaluation, while keeping retrieval or reasoning largely fixed. Mem-T (Yue et al., 2026) further connects memory formation, evolution, and multi-step retrieval by evaluating retrieval trajectories and propagating later retrieval outcomes back to earlier construction decisions. These methods move toward process-aware memory optimization, but their process feedback typically targets specific memory stages or estimates the contribution of earlier actions from later outcomes. In contrast, **AiO-Mem unifies memory interactions as transitions over a shared evidence state and uses a trajectory-conditioned reward model to directly assess the semantic quality of each memory action in a unified reward space, enabling coordinated optimization of retrieval and construction**.

C Experiment details

C.1 Datasets and baselines

Datasets The LoCoMo dataset (Maharana et al., 2024) targets the evaluation of long-range conversational memory. It features extremely long dialogues, with each conversation spanning roughly 300 turns and around 9K tokens on average. The accompanying questions cover four categories—Single-hop, Multi-hop, Temporal, and Open-domain—capturing different aspects of memory and reasoning in LLMs. Following the evaluation setup of previous work, we report F1 and BLEU-1 (B1) scores for each category as well as the overall performance.

MemoryAgentBench (Hu et al., 2026b) targets the evaluation of memory agents under incremental, multi-turn interactions. In our experiments, we evaluate three memory capabilities—Accurate Retrieval, Test-Time Learning, and Long-Range Understanding—using nine tasks from the benchmark. The selected tasks include SH-Doc QA (SH-Doc), MH-Doc QA (MH-Doc), LongMemEval (LME(S)), TREC Coarse (TREC-C), NLU, TREC Fine (TREC-F), CLINC150 (CLINC), BANKING77 (Banking), and ∞Bench-Sum (InfBench).

The PersonaMem benchmark (Jiang et al., 2025a) targets the evaluation of dynamic user profiling and personalized response generation over long-term user–LLM interactions. Unlike factual memory benchmarks that primarily focus on retrieving explicitly stated information from past interactions, PersonaMem emphasizes user-profile memory, requiring models to infer, maintain, and update users' preferences and underlying traits over time. In our experiments, we evaluate five query types corresponding to five dimensions: Recall User-Shared Facts (Rec.), Generalize to New Scenarios (Gen.), Track Full Preference Evolution (Evol.), Revisit Reasons Behind Preference Updates (Reason), and Provide Preference-Aligned Recommendations (Reco.). Together, these query types assess whether a memory system can recover user-specific facts, generalize learned preferences to new situations, track evolving preferences, reason about the causes of preference changes, and provide recommendations aligned with the user's current profile.

Baselines. We compare **AiO-Mem** against several representative baselines across three groups. ① *Reference:* Long-Context and RAG. ② *Agent Memory Harness:* MemGPT (Packer et al., 2023), MemoryBank (Zhong et al., 2024), Mem0 (Chhikara et al., 2025), MemoryOS (Kang et al., 2025), LightMem (Fang et al., 2026), A-Mem (Xu et al., 2026), GAM (Yan et al., 2025), Zep (Rasmussen et al., 2025), LangMem (LangChain, 2025), EverMemOS (Hu et al., 2026a), and Emotional RAG (Huang et al., 2024). These approaches equip a frozen backbone model with explicit memory storage, retrieval, summarization, organization,or update mechanisms. ③ *Trained Method:* MEM1 (Zhou et al., 2026), MemAgent (Yu et al., 2026a), Mem-T (Yue et al., 2026), AtomMem (Huo et al., 2026), Mem-α (Wang et al., 2025), FineMem (Ma et al., 2026), and TrustMem (Yang et al., 2026). Compared with agent memory harnesses, these baselines either learn memory-oriented agent behaviors or benefit from stronger post-trained reasoning capabilities. To ensure a fair comparison, we follow the same baseline configurations as prior work (Yue et al., 2026; Ma et al., 2026).

C.2 Implementation Details

For traceable *multi-view event bank* and *evidence-state transition modeling*, we use Qwen3-8B as the structure-aware compiler $\mathcal{C}$, dense retrieval for the summary and event-phrase views, BM25 for the query-anchor view, and a combination of BM25 and dense retrieval for the raw-turn view. We use BGE-M3[3] as the embedding model. We set the maximum interaction horizon to $T_{\max} = 8$ and the observation text budget to 10,240 tokens for evidence extraction.

For *unified reward modeling*, we construct the training data from policy rollouts. We use Qwen3-8B as the reward model. We filter low-quality and near-duplicate samples using BGE-M3 similarity, then apply agglomerative clustering (Johnson, 1967; Ward, 1963) to improve diversity and retain at most 10 representative trajectories per question. This yields 1,248 evidence-grounded SFT examples covering 150 questions and 4,162 scoreable steps, with the selected trajectories spanning diverse behavioral regions as shown in Figure 6. We derive action-specific rule scores for retrieval, note construction, citation, and stopping, prioritizing recall of missing evidence over evidence precision. We then fine-tune the reward model with LoRA SFT for 2 epochs using a global batch size of 8, a learning rate of 1×10^{-4}, and LoRA rank, alpha, and dropout of 32, 64, and 0.05. For trajectory-level DPO alignment, we use DeepSeek-V4-Flash[4] to review scoreable steps and retain cases differing from the rule-based score by one quality level, yielding 358 preference pairs from 296 trajectories and 102 questions. Human evaluation on 100 sampled steps shows strong agreement with DeepSeek-V4-Flash (ICC $= 0.87$). Finally, the reward model is further optimized with DPO for one epoch using a batch size of 8, a learning rate of 2×10^{-6}, and $\beta = 0.1$.

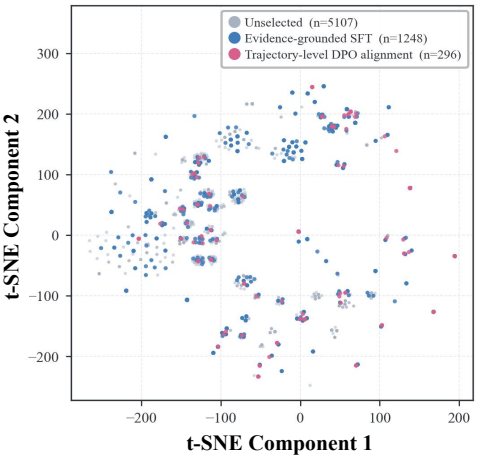


Figure 6: Reward-model trajectories by behavioral region in t-SNE space.

For *process-aligned policy optimization*, we use Qwen3-4B as the memory policy actor. We first aggregate step-level score with the action-balanced trajectory reward so that repeated `search` and `add` operations do not dominate less frequent but decisive actions such as `stop`. For reward normalization, we set $r_c^{\text{lo}} = 1$ and $r_c^{\text{hi}} = 100$ for every $c \in \mathcal{A}$. Following MEM-T (Yue et al., 2026), we adopt the same data configuration by splitting the LoCoMo dataset into training, validation, and test sets with a ratio of 1:1:8 to ensure a fair comparison. During training, we set the retrieval width to $K_{\text{ret}} = 10$, the maximum interaction horizon to $T_{\max} = 8$, and the number of rollouts to 8. The policy is optimized with LoRA using a learning rate of 2×10^{-5}, with the LoRA rank and alpha set to 32 and 64, respectively. We use Qwen3-4B as the answer model (fine-tuned on our LoCoMo training split for LoCoMo) and enable thinking mode during inference. All experiments are conducted on a single node with 8× NVIDIA A6000 GPUs (40 GB each), with a training batch size of 8. Detailed prompts are provided in Appendix G.

D Supplementary Experiments

D.1 Sensitivity Analysis

Table 3 reports the detailed F1 and B1 results for all evaluated top-K values and maximum interaction steps $T_{\max}$ on LoCoMo. Models trained with different K and $T_{\max}$ settings consistently achieve strong performance, demonstrating the generalizability of our method across training budgets. Overall F1 is highest at $K = 10, T_{\max} = 8$, which we adopt as our configuration.

[3]https://huggingface.co/BAAI/bge-m3.

[4]DeepSeek-V4-Flash refers to `DeepSeek-V4-Flash-0731`.

Table 3: Parameter analysis on LoCoMo with different top-K values and maximum interaction steps $T_{\max}$.

top-K	Max Step	Multi-hop		Temporal		Open-domain		Single-hop		Overall	
		F1	B1	F1	B1	F1	B1	F1	B1	F1	B1
$K = 5$	$T_{\max} = 4$	44.6	35.7	66.2	59.6	30.3	25.6	62.7	57.3	58.0	51.8
	$T_{\max} = 6$	43.5	35.4	62.4	53.3	34.9	30.9	65.9	61.5	59.2	53.2
	$T_{\max} = 8$	44.0	36.6	63.9	55.7	34.4	30.3	65.4	60.9	59.2	53.5
	$T_{\max} = 10$	43.0	35.1	61.9	53.1	33.5	29.3	65.8	61.1	58.8	52.8
$\mathbf{K = 10}$	$T_{\max} = 4$	43.6	35.6	63.4	54.6	33.7	30.2	65.9	60.9	59.3	53.3
	$T_{\max} = 6$	46.9	38.1	66.5	60.6	32.4	29.6	66.1	61.2	60.7	54.9
	$\mathbf{T_{\max} = 8}$	**47.1**	**38.8**	**66.0**	**59.4**	**33.7**	**30.5**	**67.0**	**61.5**	**61.0**	**55.0**
	$T_{\max} = 10$	43.5	35.3	66.1	59.2	33.0	28.9	65.9	60.1	59.8	53.4
$K = 15$	$T_{\max} = 4$	43.5	35.4	61.2	52.0	32.6	29.2	66.5	61.9	59.1	53.0
	$T_{\max} = 6$	43.5	35.7	63.3	53.6	34.8	30.9	65.8	61.0	59.3	53.0
	$T_{\max} = 8$	45.5	36.7	65.7	60.6	31.2	28.2	66.5	60.9	60.3	54.4
	$T_{\max} = 10$	45.1	36.1	65.0	58.8	30.6	26.9	65.8	60.5	59.6	53.6

D.2 Additional Results on PersonaMem

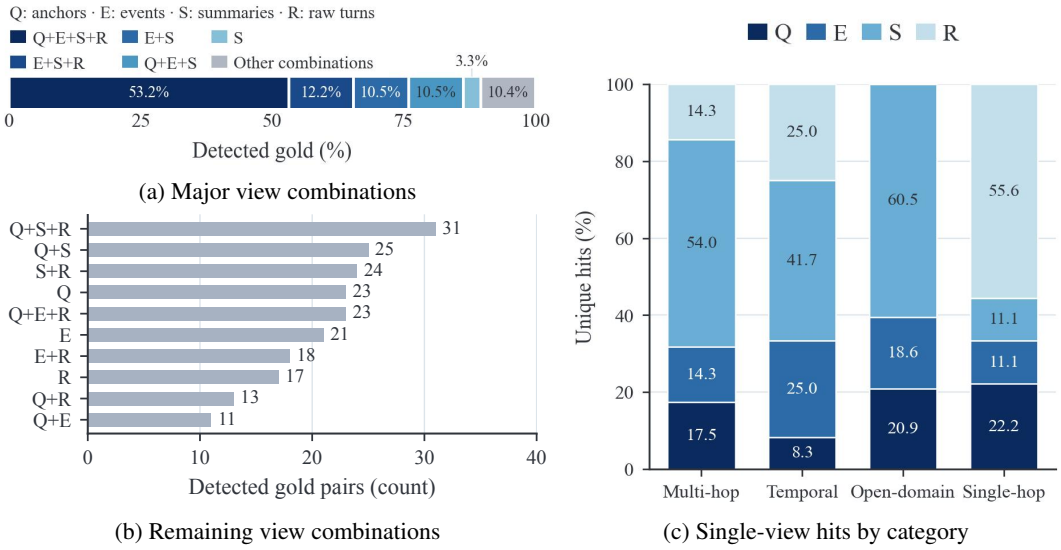
Table 4: Main results on **PersonaMem**.

Method	Rec.	Gen.	Evol.	Reason	Reco.	Overall
Claude 3.7-Sonnet	25.0	29.0	45.0	57.0	43.0	40.0
Mem0	32.1	57.9	54.7	80.8	52.7	53.4
Zep	35.3	42.1	34.5	50.5	36.4	39.1
LangMem	31.3	8.8	53.2	81.8	40.0	46.4
A-MEM	63.0	57.9	54.7	**85.9**	**69.1**	65.3
MemoryOS	47.1	56.1	32.4	48.5	47.3	44.3
MemoryBank	35.3	49.1	51.1	56.6	40.0	46.5
EverMemOS	47.1	47.4	41.0	43.4	56.4	45.7
Emotional RAG	52.9	61.4	44.6	58.6	43.6	51.6
Mem-T	61.6	61.4	56.1	72.7	45.5	60.5
AiO-Mem	**71.9**	**68.4**	**62.6**	82.8	54.6	**69.2**

We further conduct experiments on PersonaMem (Jiang et al., 2025a). It evaluates the ability of memory systems to maintain and use dynamic user profiles over long-term interactions. As shown in Table 4, AiO-Mem achieves the best overall performance with an overall score of 0.692. In contrast, existing memory systems show less balanced performance across different capabilities. A-MEM performs well on reasoning and recommendation but is weaker on generalization and preference evolution. Mem-T achieves competitive performance on recall and generalization but performs less effectively on preference-aligned recommendation. These results suggest that **the unified memory interaction in AiO-Mem enables effective retrieval, organization, and use of user-profile information from long-term interaction histories.**

D.3 Multi-View Retrieval Complementarity Study

We further analyze the complementarity of the four retrieval views using queries generated during actual inference trajectories. After merging candidates across all views, 1,990 of 2,051 question–evidence pairs are detected by at least one view, yielding 97.03% candidate-pool coverage. As shown in Figures 7a and 7b, although evidence detected by all four views forms the largest group, substantial evidence is distributed across other view combinations, demonstrating their complementary coverage. Figure 7c further shows that single-view hits vary across question categories; for example, raw turns and query anchors contribute relatively more to Single-hop questions. These results indicate that different views capture distinct evidence patterns and jointly improve coverage across diverse question types.



(a) Major view combinations

(b) Remaining view combinations

(c) Single-view hits by category

Figure 7: Complementary evidence coverage across retrieval views. Panel a summarizes the major view combinations, panel b expands the remaining combinations, and panel c shows the sources of evidence detected by exactly one view across question categories.

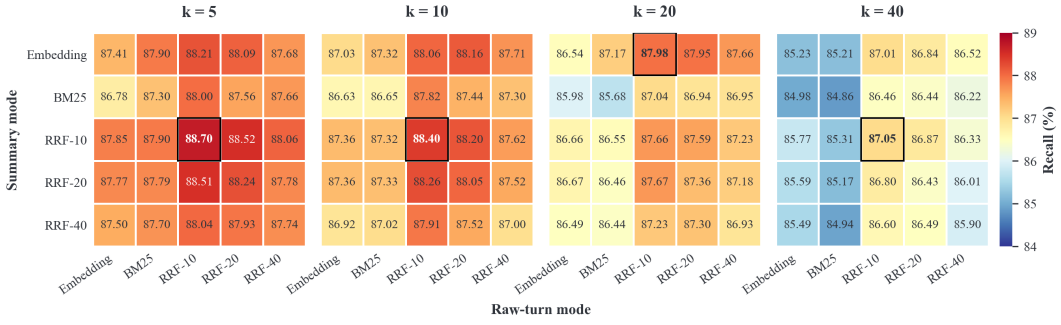
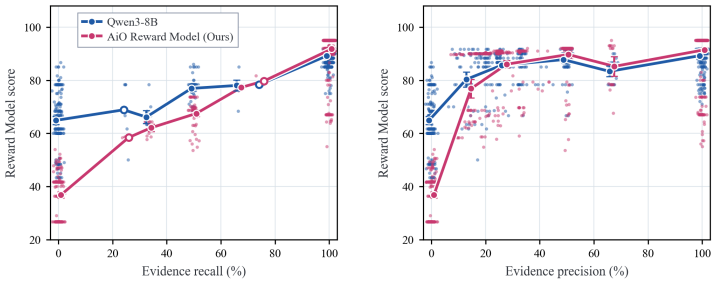


Figure 8: Sensitivity to retrieval modes and RRF parameters. Rows and columns denote the summary and raw-turn retrieval modes, respectively, while each panel uses a different aggregation RRF parameter. Red and blue indicate higher and lower evidence Recall.

Table 5: Correlations between Reward scores and evidence metrics.

	Recall		Precision	
Reward model	Pearson	Spearman	Pearson	Spearman
Qwen3-8B	0.877	0.783	0.640	0.675
AiO Reward Model (Ours)	**0.958**	**0.800**	**0.686**	**0.861**



(a) Reward vs. Recall (b) Reward vs. Precision

Figure 10: Sample-level Reward scores versus evidence metrics.

D.4 Multi-View Retrieval Sensitivity Study

The retrieval-budget analysis in Figure 9 characterizes the trade-off between evidence coverage and context cost, while Figure 8 examines sensitivity to retrieval configurations. As K_{ret} increases from 5 to 30, evidence Recall improves steadily from 73.62% to 90.82%, while the average retrieved context grows substantially from 2,197 to 11,137 tokens. At $K_{\text{ret}} = 20$, the system already achieves 88.16% Recall with 7,743 tokens; increasing the budget to 30 brings only a further 2.66-point Recall gain at approximately 44% higher context cost, indicating diminishing returns from larger retrieval budgets. We therefore set $K_{\text{ret}} = 20$ as a practical balance between evidence coverage and context efficiency. Figure 8 further shows that moderate variations around the default retrieval modes and RRF parameters cause only limited changes in Recall, whereas larger deviations can lead to noticeable degradation. This suggests that the adopted retrieval configuration lies in a relatively stable and high-performing region rather than depending on a narrowly tuned parameter setting.

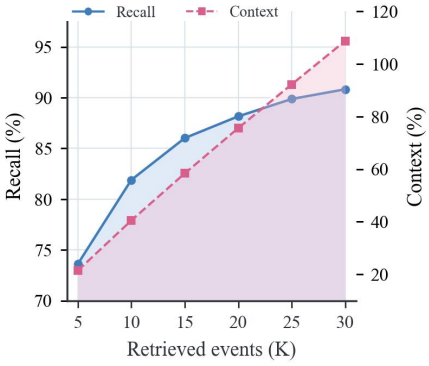


Figure 9: Trade-off between retrieval Recall and context cost.

E Reward and Optimization Analysis

E.1 Unified Reward Model Alignment Study

We further evaluate whether the trained reward model aligns with evidence usage by measuring its correlation with trajectory-level Recall and Precision. As shown in Table 5, training increases the Pearson correlation with Recall from 0.877 to 0.958 and the Spearman correlation with Precision from 0.675 to 0.861, indicating stronger alignment with evidence-based trajectory quality. Figures 9a and 9b further show that the base reward model assigns relatively concentrated scores across different evidence qualities, whereas the trained model exhibits clearer positive trends with both Recall and Precision. Notably, some trajectories still receive high rewards at different Precision levels, suggesting that the trained reward model captures broader trajectory quality beyond evidence Precision alone.

E.2 TRAJECTORY-LEVEL DPO ALIGNMENT STUDY

Evidence-grounded SFT provides action-level supervision based on evidence coverage and provenance, but such rule-based signals cannot fully assess the semantic quality of an action within its trajectory context. As introduced in Section 3.4, trajectory-level DPO addresses this limitation by using full trajectory context to align the reward model with finer-grained semantic judgments. To visualize how trajectory-level DPO distinguishes trajectories beyond rule-based evidence matching, we compare four matched trajectory pairs for the same questions. Within each pair, the trajectories have identical Evidence-ID Recall and Precision but receive different scores after trajectory-level DPO alignment.

Specifically, in q114, both trajectories cite the correct evidence ID, but the lower-Reward trajectory mixes visual details from another event into the adoption meeting, whereas the higher-Reward trajectory remains focused on the target event. In q76, both obtain zero Evidence-ID Recall and Precision, yet the higher-Reward trajectory retains evidence supporting "three children," while the lower-Reward one infers "two children" from insufficient observations. In q65, despite complete evidence coverage, the lower-Reward trajectory repeats the same fact and conflates Melanie's stated preference with the inference about Vivaldi, whereas the higher-Reward trajectory separates the source fact from the inference. In q7, the lower-Reward trajectory mixes the target future camping plan with a past camping event, while the higher-Reward trajectory preserves the correct temporal focus. These cases show that trajectory-level DPO complements rule-based supervision by aligning the reward model with evidence attribution, relevance, factual organization, temporal consistency, and redundancy beyond Evidence-ID matching.

Matched Trajectories for q114

Question: What did Caroline see at the council meeting for adoption?

Gold answer: Many people wanting to create loving homes for children in need.

Gold evidence: `D8:9`

Both trajectories: Evidence-ID Recall = **1.00**; Precision = **1.00**.

Lower Trajectory-level Reward Model Score **Score: 0.67**

Case annotation. The notes retain the relevant observation about prospective adoptive families, but also attach visual details from another event to the same meeting. Both statements cite the gold evidence ID, concealing the mismatch between the cited source and the additional claim.

- Caroline described the adoption council meeting as inspiring and emotional, noting that many people wanted to create loving homes for children in need. *(Evidence: `D8:9`)*
- Caroline saw **a sign for a new arrival and an information and domestic building** during the adoption council meeting. *(Evidence: `D8:9`)*

Higher Trajectory-level Reward Model Score **Score: 0.95**

Case annotation. The notes consistently describe the adoption meeting and the people seeking to provide loving homes. Although the two entries overlap, both remain focused on what Caroline observed at the target event, without introducing the unrelated visual details.

- Caroline attended an adoption council meeting and saw **people seeking to create loving homes for children in need**. *(Evidence: `D8:9`)*
- Caroline described the adoption council meeting as inspiring and emotional, with attendees focused on **creating loving homes for children in need**. *(Evidence: `D8:9`)*

Matched Trajectories for q76

Question: How many children does Melanie have?

Gold answer: Three.

Gold evidence: `D18:1, D18:7`

Both trajectories: Evidence-ID Recall = **0.00**; Precision = **0.00**.

Lower Trajectory-level Reward Model Score **Score: 0.27**

Case annotation. The notes use observations of two children in particular scenes to support a claim about Melanie's total number of children. Seeing two children at a campfire or museum does not establish that she has only two, leaving the numerical conclusion insufficiently supported.

- Melanie **has two children based on the image of a man and two children** in the campfire scene. *(Evidence: `D8:32`)*
- Melanie took two children to the museum. *(Evidence: `D6:4`)*

Higher Trajectory-level Reward Model Score **Score: 0.53**

Case annotation. The notes preserve an observation involving three children and additional family context. Neither entry matches the annotated gold evidence IDs, but the beach observation is more relevant to the reference answer. The displayed notes alone still do not conclusively establish the total number of Melanie's children.

- Melanie went to the beach with **three children**. *(Evidence: `D10:8`)*
- Melanie's children were at the Grand Canyon with her. *(Evidence: `D18:5`)*

Matched Trajectories for q65

Question: Would Melanie likely enjoy the song "The Four Seasons" by Vivaldi?

Gold answer: Yes; it is classical music.

Gold evidence: `D15:28`

Both trajectories: Evidence-ID Recall = **1.00**; Precision = **1.00**.

Lower Trajectory-level Reward Model Score **Score: 0.55**

Case annotation. The notes repeat Melanie's classical music preference and conflate it with an inferred preference for Vivaldi.

- Melanie enjoys classical music including composers like Bach and Mozart. *(Evidence: `D15:28`)*
- **Melanie enjoys classical music including composers like Bach and Mozart.** *(Evidence: `D15:28`)*
- Melanie appreciates classical music genres. *(Evidence: `D15:28`)*
- Melanie appreciates classical music genres including composers like Bach, Mozart, and **implicitly Vivaldi**. *(Evidence: `D15:28`)*

Higher Trajectory-level Reward Model Score **Score: 0.95**

Case annotation. The notes distinguish Melanie's stated preference, the genre connection, and the inference that she would likely enjoy Vivaldi.

- Melanie enjoys **classical music by Bach and Mozart**, as well as modern music like Ed Sheeran's "Perfect". *(Evidence: `D15:28`)*
- Vivaldi's "The Four Seasons" is **a classical composition**, and Melanie appreciates classical music by Bach and Mozart. *(Evidence: `D15:28`)*
- Melanie's appreciation for classical music **suggests she would likely enjoy** Vivaldi's "The Four Seasons". *(Evidence: `D15:28`)*

Matched Trajectories for q7

Question: When is Melanie planning on going camping?

Gold answer: June 2023.

Gold evidence: `D2:7`

Both trajectories: Evidence-ID Recall = **1.00**; Precision = **0.50**.

Lower Trajectory-level Reward Model Score **Score: 0.54**

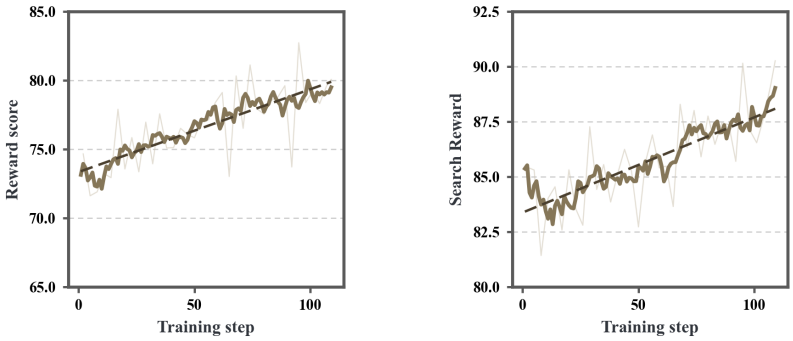
Case annotation. The notes repeatedly alternate between a future camping plan and a separate camping trip that had already occurred. Although the relevant expression "next month" eventually appears, repetition and the interleaving of different events make the target plan less prominent.

- Melanie plans a camping trip for summer break. *(Evidence: `D2:7`)*
- Melanie **went camping with kids a few weeks ago**. *(Evidence: `D16:2`)*
- **Melanie plans a camping trip for summer break.** *(Evidence: `D2:7`)*
- Melanie **went camping with kids a few weeks ago**. *(Evidence: `D16:2`)*
- Melanie plans a camping trip for next month during summer break. *(Evidence: `D2:7`)*

Higher Trajectory-level Reward Model Score **Score: 0.92**

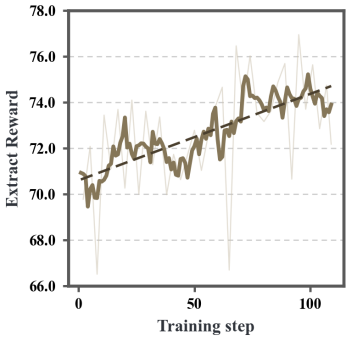
Case annotation. The notes foreground the upcoming trip and its time expression, "next month," followed by broader summer-outing context. They avoid repeatedly mixing the plan with the completed camping event, making the information relevant to the temporal question more accessible.

- Melanie mentioned **planning a camping trip for next month** during summer break. *(Evidence: `D2:7`)*
- Melanie agreed to plan a family outing this summer, which includes camping activities. *(Evidence: `D12:18`)*

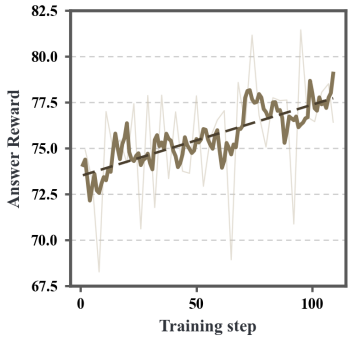


(a) Overall Reward

(b) Search Reward



(c) Extract Reward



(d) Answer-Decision Reward

Figure 11: Training dynamics.

E.3 Training Dynamics

Figure 11 presents the Reward dynamics during PA-DAPO training. As shown in Figure 11a, the Overall Reward exhibits a clear upward trend as training progresses. Although the Reward varies across individual training steps, the smoothed curve increases overall, indicating a gradual improvement in the quality of the sampled trajectories.

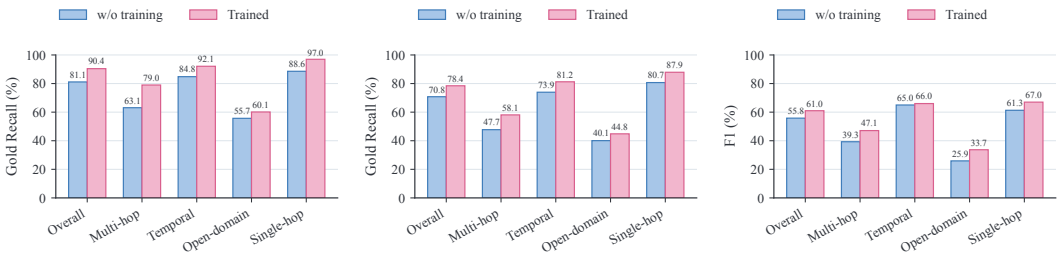
Figures 11b–11d further decompose the Overall Reward by operation: Search corresponds to search, Extract includes add and skip, and Answer-Decision corresponds to stop.The Search Reward shows an overall increase, indicating improved evidence retrieval behavior. The Extract Reward also increases over training, reflecting improvements in evidence extraction and organization. The Answer-Decision Reward exhibits a similar upward trend, indicating that the model progressively learns when to stop retrieving and proceed to answer generation.

The overall improvement across all three stages indicates that the increase in Overall Reward is not driven by a single interaction stage. Instead, it reflects joint improvements in evidence retrieval, evidence extraction and organization, and the decision to stop retrieving and generate an answer.

E.4 Process-Aligned Evidence Flow Study

We first analyze how training affects evidence flow through the memory pipeline. Figures 12a, 12b, and 12c compare the trained and untrained systems at the Search, Final working memory, and Answer stages, respectively, with further breakdowns by question category.

After training, the system covers more gold evidence during Search and retains more evidence in the Final working memory, with improvements across all question categories. The final Answer F1 also increases accordingly. These results show that policy training improves the entire information processing pipeline, from evidence retrieval and organization to answer generation.



(a) Search (b) Final working memory (c) Answer

Figure 12: Effects of policy training on evidence flow through the memory pipeline. The three panels report Search Recall, Final working memory Recall, and Answer F1, respectively.

F Computational Complexity

Table 6: Complexity comparison between AiO-Mem and OMS.

System	(I) Segment & Retrieval	(II) Construction	(III) Inference
OMS	$O(N)\cdot\mathcal{C}_{\text{seg}}$ $+O(N\ell)\cdot\mathcal{C}_{\text{LLM}}$ $+O(NM_1)\cdot\mathcal{C}_{\text{emb}}$	$O(NM_1)\cdot\mathcal{C}_{\text{sim}}$ $+O(NM_1R_1)\cdot(\mathcal{C}_{\text{LLM}}+\mathcal{C}_{\text{emb}})$	$O(N\ell+NM_1R_1)\cdot\mathcal{C}_{\text{LLM}}$ $+O(NM_1(1+R_1))\cdot\mathcal{C}_{\text{emb}}$ $+O(NM_1)\cdot\mathcal{C}_{\text{sim}}$
AiO-Mem	$O(T_{\max}M)\cdot\mathcal{C}_{\text{sim}}$ $+O(N\ell)\cdot\mathcal{C}_{\text{LLM}}$ $+O(M+T_{\max})\cdot\mathcal{C}_{\text{emb}}$	$O(T_{\max}K_{\text{ret}}\ell_e)\cdot\mathcal{C}_{\text{LLM}}$	$O(N\ell+T_{\max}K_{\text{ret}}\ell_e)\cdot\mathcal{C}_{\text{LLM}}$ $+O(M+T_{\max})\cdot\mathcal{C}_{\text{emb}}$ $+O(T_{\max}M)\cdot\mathcal{C}_{\text{sim}}$

As shown in Table 6, we analyze a long-horizon history $\mathcal{H}=(h_1,\ldots,h_N)$ consisting of N source units, each containing an average of ℓ tokens, and evaluate the cost of answering a memory query q. In other memory systems (OMS), the history is first segmented into source units, where each unit requires an LLM-based summarization and is further encoded into M_1 persistent memory entries:

$$\mathcal{C}^{\text{I}}_{\text{OMS}}=O(N)\cdot\mathcal{C}_{\text{seg}}+O(N\ell)\cdot\mathcal{C}_{\text{LLM}}+O(NM_1)\cdot\mathcal{C}_{\text{emb}}. \tag{6}$$

During memory construction, the resulting NM_1 entries are compared with the existing memory. A fraction R_1 of these entries triggers an LLM-based rewrite, followed by re-embedding for index update:

$$\mathcal{C}^{\text{II}}_{\text{OMS}}=O(NM_1)\cdot\mathcal{C}_{\text{sim}}+O(NM_1R_1)\cdot(\mathcal{C}_{\text{LLM}}+\mathcal{C}_{\text{emb}})\,. \tag{7}$$

This results in a total inference cost of:

$$\begin{aligned}\mathcal{C}^{\text{infer}}_{\text{OMS}}=O(N\ell+NM_1R_1)\cdot\mathcal{C}_{\text{LLM}}\\ +O(NM_1(1+R_1))\cdot\mathcal{C}_{\text{emb}}+O(NM_1)\cdot\mathcal{C}_{\text{sim}}.\end{aligned} \tag{8}$$

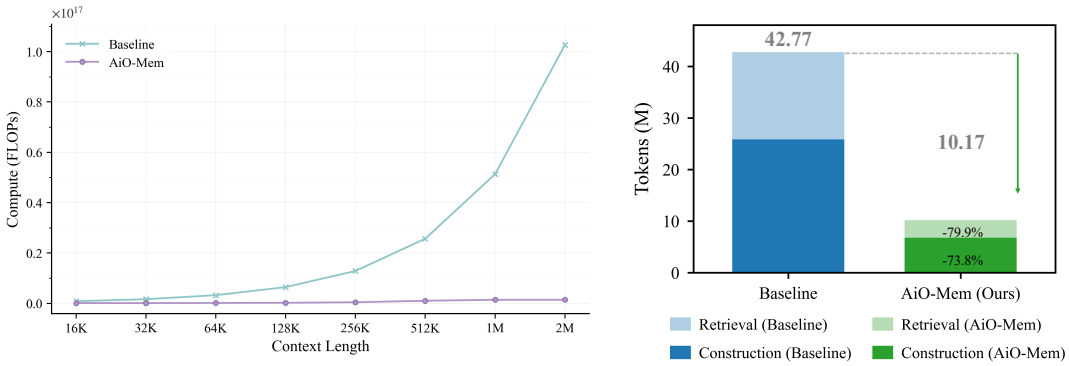
Thus, both the LLM-based summarization and memory rewriting costs increase with the history size and the number of accumulated memory entries.

In **AiO-Mem**, the history is first compiled into the traceable event bank $\mathcal{B}=\mathcal{C}(\mathcal{H})$. The resulting M events and their indices are reused across queries. Since each trajectory contains at most $T_{\max}$ interaction steps, the segmentation and retrieval cost is:

$$\mathcal{C}^{\text{I}}_{\text{AiO}}=O(N\ell)\cdot\mathcal{C}_{\text{LLM}}+O(M+T_{\max})\cdot\mathcal{C}_{\text{emb}}+O(T_{\max}M)\cdot\mathcal{C}_{\text{sim}}. \tag{9}$$

On top of this, AiO-Mem keeps $\mathcal{B}$ unchanged and constructs only the query-specific working memory $\mathcal{W}_t$. Each interaction step operates on at most K_{ret} retrieved events with an average exposed length of ℓ_e tokens. Therefore, the construction cost is:

$$\mathcal{C}^{\text{II}}_{\text{AiO}}=O(T_{\max}K_{\text{ret}}\ell_e)\cdot\mathcal{C}_{\text{LLM}}. \tag{10}$$



(a) FLOPs comparison on LoCoMo.

(b) Token comparison on LoCoMo.

Figure 13: Efficiency comparison between AiO-Mem and Baseline on LoCoMo.

Table 7: Notation used in the complexity analysis.

Symbol	Definition
$\mathcal{H} = (h_1, \ldots, h_N)$	Long-horizon history consisting of N heterogeneous source units.
N	Number of source units in $\mathcal{H}$.
ℓ	Average number of tokens in each source unit h_n.
$\mathcal{B} = \mathcal{C}(\mathcal{H})$	Traceable event bank constructed from $\mathcal{H}$ by the structure-aware compiler $\mathcal{C}$.
M	Number of traceable events in the event bank $\mathcal{B}$.
M_1	Number of persistent memory entries produced per source unit by OMS.
R_1	Fraction of OMS memory entries that trigger memory rewriting and re-indexing.
K_{ret}	Number of candidate events returned by each SEARCH action.
ℓ_e	Average number of tokens exposed by one retrieved event.
$T_{\max}$	Maximum interaction horizon, with $T \le T_{\max}$.
$\mathcal{C}_{\text{seg}}$	Cost of one segmentation operation.
$\mathcal{C}_{\text{LLM}}$	Cost of LLM computation.
$\mathcal{C}_{\text{emb}}$	Cost of an embedding operation.
$\mathcal{C}_{\text{sim}}$	Cost of a sparse or dense similarity operation.

Combining event-bank construction, multi-view retrieval, bounded working-memory construction, and final response generation, AiO-Mem requires

$$\begin{aligned} \mathcal{C}_{\text{AiO}}^{\text{infer}} &= O(N\ell + T_{\max} K_{\text{ret}} \ell_e) \cdot \mathcal{C}_{\text{LLM}} \\ &\quad + O(M + T_{\max}) \cdot \mathcal{C}_{\text{emb}} + O(T_{\max} M) \cdot \mathcal{C}_{\text{sim}}. \end{aligned} \tag{11}$$

Following prior efficiency-oriented works, we report FLOPs measured by (Ye, 2023) and token consumption, using Mem-T as the baseline. As shown in Figure 13, AiO-Mem exhibits substantially slower FLOPs growth with increasing context length. By decoupling large-scale persistent memory management from query-time working-memory construction, AiO-Mem avoids repeated memory rewriting and re-indexing and confines costly LLM computation to selected evidence. It further reduces retrieval and construction tokens by 79.9% and 73.8%, respectively.

G PROMPTS AND TOOL CALLS

G.1 MEMORY AGENT TOOL CALLS

The LoCoMo memory agent alternates between DECIDE and EXTRACT. DECIDE chooses between search and stop; after each search, EXTRACT chooses between add, which writes visible evidence into working memory, and skip.Table 8 lists the tools available at each stage.

Table 8: Tool calls available to the memory agent.

Stage	Tool call and function
DECIDE	`search_memory_units(query)`: Retrieves event-level memory units concerning information missing from the current notes.
DECIDE	`answer_from_notes()` (no arguments): Ends memory retrieval and proceeds to the answering stage using the existing working memory.
EXTRACT	`add_note(event_id, datetime, content, evidence_turns)`: Saves a clue, its event time, and the original dialogue supporting it. An EXTRACT step may invoke `add_note` multiple times, with each invocation writing one note.
EXTRACT	`skip_extraction(reason)`: Skips extraction when the current search results contain no relevant evidence to write.

G.2 LOCOMO MEMORY AGENT PROMPTS

Table 9 presents condensed prompt templates for the two model-call stages. Each call includes only the tool definitions permitted at that stage; the model returns tool calls as JSON objects enclosed in `<tool_call>` tags.

Table 9: Prompt templates for the LoCoMo memory agent.

Stage	Prompt template
DECIDE	You are a memory assistant. Given the current question and saved working memory, choose the next action: search memory, or stop retrieval and proceed to the answering stage. **Rules:** (1) Judge whether the question can be answered completely using only the working memory, including the relationships and all necessary parts requested by the question. If the evidence is sufficient, call `answer_from_notes`. (2) Otherwise, generate a query targeting one specific fact or clue that is still missing, and call `search_memory_units`. The query should identify the relevant person or object, relationship, and information needed. Avoid repeating previous searches that produced no useful notes. (3) At the final decision step, call `answer_from_notes`. **Input:** Question {question}, working memory {working_notes}, previous queries {previous_searches}, and step {step}/{max_steps}. **Output:** A brief explanation of the decision and one `<tool_call>`.
EXTRACT	You are a memory system analyst. Given the question, the latest search results, and the existing working memory, identify new evidence useful for answering the question and write it into the working memory. Search results are temporary; useful information not written into the notes will not be available to subsequent decisions. **Rules:** (1) Read the visible original dialogue in retrieved events. Highlighted turns are only hints; relevant evidence may also appear in other turns. (2) For each new useful clue, call `add_note` to write a concise, faithful note, preserving the exact event ID, recorded time, and original dialogue supporting its content. Each note should capture one natural clue and should be compared with existing notes to avoid duplication. (3) If the search results contain no relevant evidence, call `skip_extraction`. **Input:** Question {question}, search observation {search_observation}, and working memory {working_notes}. The search observation contains event identifiers, recorded times, retrieved fields, summaries, and visible original dialogue. **Output:** A brief assessment of the evidence and the required `<tool_call>`; distinct clues may be written as separate notes.

G.3 STEPWISE REWARD MODEL PROMPT

For SFT, the prompt below is the reward model input, with rule-derived scores as targets. For DPO, the same prompt is given to both the reward model and DeepSeek-V4-Flash; DeepSeek's score sequence serves as chosen and the rule-derived sequence as rejected. Reference answers and annotated evidence are used only for reward scoring, not by the memory policy or answer model. Table 10 summarizes the prompt's objectives, error categories, score ranges, and output format.

Table 10: Prompt template and scoring rules for the stepwise reward model.

Prompt template

Task and input
You are a reward model evaluator of memory trajectories. Given a complete scoreable memory trajectory, evaluate each step according to whether the required evidence was found, whether useful evidence was written into working memory, and whether retrieval stopped when the evidence was sufficient. You may use trajectory context to understand the current action, but each score must evaluate only the current step; do not evaluate the quality of the final answer model's response.
Input: {question}, {reference_answer}, {gold_evidence}, {trajectory_information}, and {scoreable_steps}.
Objectives by action type
`decide_search`: Does the query target information missing from the current notes and help locate evidence needed to answer the question? A reasonable query should not receive the lowest score solely because it failed to retrieve a match.
`extract_add_note`: Does the action faithfully write useful, previously unsaved evidence from the latest search into the notes, and do the cited turns support the note's content?
`extract_skip`: Does the latest search truly contain nothing worth writing? Skipping clearly critical evidence is a severe problem.
`decide_answer_from_notes`: Are the existing notes sufficient to support an answer when retrieval stops?
Evidence assessment
When judging whether the notes are sufficient, consider both their content and the original dialogue they cite. Annotated evidence indicates information needed for the question, but it is not a mechanical hit-rate scoring formula: unannotated dialogue turns may also support the answer semantically, and matching annotated turns does not automatically warrant a high score.
Error severity
Category A, high severity: Failure of the current stage's core objective.
Category B, high severity: Factual or attribution errors in a note, or citations that do not support key content.
Category C, medium severity: Problems in assessing the current state or respecting stage boundaries.
Category D, medium severity: An incomplete contribution or substantial noise in the action.
Category E, low severity: Wording or formatting issues.
Score ranges
90–100: The action fulfills the stage's core objective very well, with at most minor Category E issues.
75–89: The action is clearly useful and largely fulfills the core objective; it has no Category A or B issues, though minor Category C, D, or E issues are allowed.
60–74: The action makes a partial contribution but falls short of the core objective; it has no Category A or B issues, though Category C or D issues are more pronounced.
40–59: The action has a severe problem but retains some useful contribution; it typically contains one Category A or B issue.
1–39: The action provides almost no useful contribution or takes a clearly wrong direction; severe problems cause the current step to largely fail.
Output requirements
Output a strict JSON object containing `step_scores` in input order. Each step must contain `step_index`, `step_type`, `quality_level`, and an integer `score` from 1 to 100. Do not omit or add steps, and do not output scoring explanations.
`quality_level` roughly corresponds to the score as follows: `bad` (1–30), `weak` (31–55), `partial` (56–70), `good` (71–85), and `excellent` (86–100).