MemCoRL: Alternating Co-Optimization of Memory Retrieval and Utilization via Collaborative Reinforcement Learning

Yuewen Liu, Peng Xu*, Muxi Diao, Anyi Zhang, Yang Li, Yutong Zhang
State Key Lab of Networking and Switching Technology
Beijing University of Posts and Telecommunications
{liuyw,xupeng,dmx}@bupt.edu.cn

Abstract

Large Language Models (LLMs) are inherently constrained by their fixed-length context windows, which limits LLMs' ability to retain and utilize information across long-term interactions. To address this limitation, recent work has proposed external memory modules for LLMs. Using memory modules typically involves two stages: evidence retrieval and memory utilization. While prior work focuses on the architecture of memory modules and the retrieval stage, the equally critical memory utilization stage remains underexplored. Building on this, we propose MemCoRL, a two-stage alternating co-optimization reinforcement learning method. Stage 1 optimizes evidence retrieval using citation feedback and semantic accuracy from utilization as rewards. Stage 2 optimizes utilization with rewards combining semantic similarity and lexical overlap. Iterative co-optimization establishes a positive feedback loop: better retrieval improves memory utilization, which in turn refines retrieval rewards. Experimental results show our approach outperforms the leading baselines on both lexical overlap and semantic similarity metrics, confirming the co-optimization in memory retrieval and memory utilization.

1 Introduction

Large Language Models (LLMs) have demonstrated impressive capabilities in text generation and dialogue tasks (Yang et al., 2025; OpenAI, 2023). When faced with long-term tasks that require continuous interaction, the need for memory management in LLMs becomes evident (Zhang et al., 2025). Yet LLMs' fixed-length context windows impose strict limits on memory management, making it difficult to sustain long-horizon interactions. Although many studies attempt to extend the input context of LLMs(Peng et al., 2023; Ding et al., 2024), they suffer performance degradation as the window expands and remain fundamentally constrained by input length limits(Li et al., 2024).

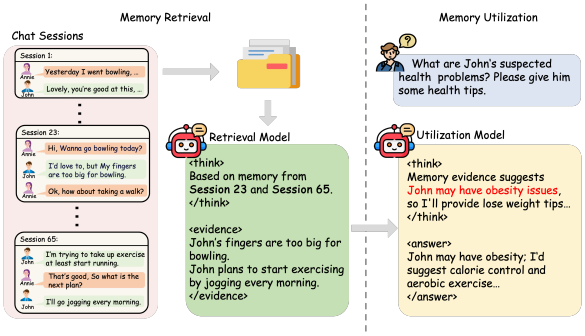

Figure 1: MemCoRL retrieves query-relevant information from extensive historical context, consolidates it into memory evidences, and utilizes them to inform inference during response generation.

To address this challenge, existing research has proposed a series of memory mechanisms that enable LLMs to overcome the limitations of finite input length and obtain contextual memory beyond the current input. These can be categorized into the following types: Knowledge organization methods, such as A-Mem(Xu et al., 2025) and Mem0 (Chhikara et al., 2025), achieve efficient memory management by designing interconnected memory structures and extracting and integrating knowledge bases. Architecture-level solutions, such as MemGPT(Packer et al., 2024) and MemoryOS(Kang et al., 2025), draw on operating-system and memory-hierarchy mechanisms to devise distinctive memory-management architectures that permit more rational and efficient handling of memories. Retrieval-centric approaches, in addition to research on optimizing the retrieval pipeline, such as DeepRAG(Guan et al., 2025), RMM(Tan et al., 2025), iteratively refine the retrieval with online Reinforcement Learning (RL) based on LLMs' cited evidence.

Existing studies emphasize external memory mechanisms for LLMs, encompassing memory rep-

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 38912–38924
July 2-7, 2026 ©2026 Association for Computational Linguistics

resentations, management strategies, and retrieval techniques. Although such refinements enhance memory formation and retrieval, they neglect a vital dimension: memory utilization referring to how LLMs employ retrieved memory to address current tasks and critically influences system performance by underpinning the reasoning processes that leverage memory across tasks (Liu et al., 2025).

Previous studies indicate that RL can independently improve reasoning and retrieval capabilities(Guo et al., 2025; Tan et al., 2025).Furthermore, we noticed that retrieval outputs directly inform utilization, while utilization performance reflects retrieval quality (Shao et al., 2023). However, effectively leveraging this reciprocal relationship to jointly enhance memory retrieval and utilization remains a challenge. To address this, we propose MemCoRL: a two-stage, alternating co-optimization framework with collaborative RL, co-optimizing memory retrieval and utilization policies. Specifically, our approach consists of two stages. Stage I: Memory Retrieval Optimization — the retrieval policy is optimized by citation feedback from the utilization policy, supplemented by a semantic reward to mitigate reward hacking(Shihab et al., 2025). Stage II: Memory Utilization Optimization — the utilization policy is optimized by a composite reward that linearly combines semantic and lexical. Through iterative training, enhanced retrieval provides precise evidence for utilization, whose refined rewards subsequently guide retrieval optimization. The retrieval and utilization policies function as interdependent sub-policy; by alternately optimizing them, MemCoRL leverages co-evolutionary dynamics whereby improved utilization feedback reshapes retrieval rewards to escape local optima, and enhanced retrieval evidence expands the utilization policy's exploration space to avoid stagnation. This co-optimization allows MemCoRL to converge on a more consistent global optimum than either policy optimized in isolation.

We evaluated our system on diverse benchmarks, including long-term conversational memory and contextual memory tasks, where it outperforms current leading baselines in lexical-level metrics (F1 and BLEU) and semantic-level metrics (LLM-as-judge), which demonstrates that our approach effectively enables the alternating co-optimization of memory evidence retrieval and memory utilization.

In conclusion, our contributions can be summarized as follows:

- We refine the reward function in existing memory retrieval methods and employ RL to optimize the memory utilization policy. Experimental results validate the efficacy of our optimization with RL.
- We focus on the reciprocal interaction between memory retrieval and memory utilization, and introduce MemCoRL, an alternating co-optimization framework based on collaborative RL, which enables the co-evolution of both the retrieval and utilization policies.
- We conduct experiments on diverse datasets. Experimental result shows MemCoRL outperforms leading baselines in F1, BLEU, and LLM-as-judge, validating the effectiveness of our approach.

2 Related Work

2.1 Memory Module for LLMs

In recent years, a line of researches focus on explicit knowledge organization. A-Mem constructs an interconnected note structure, leveraging semantic links to enable richer associative retrieval (Xu et al., 2025). Mem0 introduces graph-based representations to capture complex relationships among conversational elements and supports a scalable extract–integrate–retrieve pipeline (Chhikara et al., 2025). These approaches excel in structuring and semantically linking memories. Another direction operates at the system-architecture level. MemGPT draws on hierarchical storage in the operating system to separate a finite 'primary context' from an unbounded 'external context', enabling the model to page and retrieve memory via function calls (Packer et al., 2023). However these two strategies do not explore memory retrieval and utilization within the model itself. While traditional RAG systems retrieve from static external corpora, memory retrieval draws from interaction history—more task-relevant but also more prone to confusion. Consequently, standard RAG approaches cannot be directly applied to memory tasks. Retrieval-oriented research in memory has therefore explored incorporating reinforcement learning into the retrieval process. RMM applies online RL to iteratively align the retriever with evidence cited by LLMs during generation.(Tan et al., 2025) While effective, it updates only the retriever and excludes memory utilization from the joint training loop.

2.2 Reinforcement Learning for LLMs

RL has significantly enhanced LLMs' reasoning capabilities. Especially, Group Relative Policy Optimization(GRPO) operates without the need for a value network, directly computing advantages by normalizing rewards within the group, demonstrating excellent performance and reduced resource consumption(Shao et al., 2024)(Diao et al., 2026). As a result, the reward signal progressively transitions from human preferences(Ouyang et al., 2022) or (Bai et al., 2022) reward models to a rule-based mechanism. Beyond conventional reasoning tasks, such as coding or mathematical problem solving, existing studies have also verified the effectiveness of RL in other contexts, including search (Jin et al., 2025) and agent (Ouyang et al., 2025), in boosting model capabilities. These studies provide possibilities for enhancing LLM performance in memory-related tasks. Nevertheless, applying RL to memory scenarios still faces the challenge of clearly defining tasks to construct an effective reward mechanism.

3 MemCoRL

In this section, we will introduce our approach MemCoRL and detail its training design in stages.

3.1 Preliminaries

3.1.1 Reinforcement Learning for LLMs.

The token generation process of LLMs can be formulated as a token-level Markov decision process (MDP), in which the action space A corresponds to the model's vocabulary, with each token representing an action. Consequently, let the LLMs serve as the policy model, let V represent a finite vocabulary of tokens. The model π_θ, the policy model, takes an input prompt $x \in X$ and generates a distribution over an answer $y \in Y$, where X and $Y \subset V^*$ are the sets of possible input prompts and output sequences, respectively. Given a reward model (or function) r, the policy π_θ is optimized through policy gradient methods, with the optimization object being to maximize the expected reward while incorporating KL regularization:

$$\max_\theta \ \mathcal{J}(\theta) = \mathbb{E}_{\substack{x\sim\mathcal{D} \\ y\sim\pi_\theta(\cdot|x)}} \Big[r_\phi(x,y) - \beta\, D_{\mathrm{KL}}\big(\pi_\theta(y \mid x) \,\|\, \pi_{\mathrm{ref}}(y \mid x)\big)\Big]. \tag{1}$$

Popular policy gradient methods include PPO (Schulman et al., 2017), RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), and Reinforce++ (Hu et al., 2025).

3.1.2 Group Relative Policy Optimization (GRPO)

has demonstrated simplicity and efficacy in RL by dispensing with a critic network typically as large as the policy model and instead deriving the baseline term from group-level performance. Concretely, for each input query q, GRPO samples a set of candidate outputs $\{o_1, o_2, \ldots, o_G\}$ from the previous policy $\pi_{\theta_{\text{old}}}$, and then optimizes the policy model π_θ by maximizing the following objective:(Shao et al., 2024)

$$\begin{aligned}\mathcal{J}_{GRPO}(\theta) = \mathbb{E}_{q\sim P(Q),\,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(O|q)} \\ \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\Big\{\min\Big[\frac{\pi_\theta(o_{i,t}\mid q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid q,o_{i,<t})}\hat{A}_{i,t}, \\ \text{clip}\Big(\frac{\pi_\theta(o_{i,t}\mid q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid q,o_{i,<t})}, 1-\epsilon,\, 1+\epsilon\Big)\hat{A}_{i,t}\Big] \\ -\beta\, D_{\mathrm{KL}}\big[\pi_\theta \,\|\, \pi_{\mathrm{ref}}\big]\Big\}.\end{aligned} \tag{2}$$

Where ε and β are hyperparameters, and the advantage A_i is calculated from a group of rewards $\{r_1, r_2, \ldots, r_G\}$ as follows:(Guo et al., 2025)

$$A_i = \frac{r_i - \text{mean}\left(\{r_1, r_2, \ldots, r_G\}\right)}{\text{std}\left(\{r_1, r_2, \ldots, r_G\}\right)}. \tag{3}$$

GRPO has been extensively validated to enhance performance across diverse tasks (Sun et al., 2025; Wei et al., 2025); accordingly, we integrate it into our algorithmic framework to compute advantage estimates and update policy models.

3.2 Method Overview

Our framework consists of two components: the Memory Retrieval Policy π_r and the Memory Utilization Policy π_u, which operate sequentially and reinforce each other. Given a query q, π_r retrieves relevant memory M from context C, which is used by π_u to generate reasoning and an answer 1. We train both policies in two stages: memory retrieval optimization and memory utilization optimization.

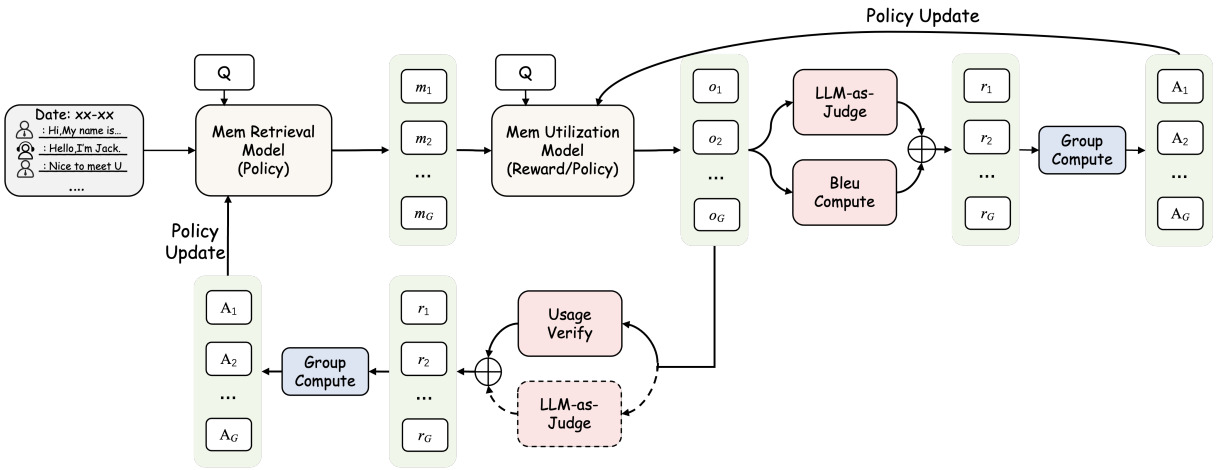

Figure 2: Overview of MemCoRL, a collaborative RL framework for co-optimizing memory retrieval and utilization. The framework combines group-relative policy optimization, hybrid reward mechanisms, and alternating co-optimization between retrieval and utilization.

3.3 Stage1: Memory Retrieval Optimization

The task of this stage is to retrieve memory evidence from the historical context C, based on the current query q. We employ LLMs to first extract all information relevant to query q, followed by selection and integration, ultimately summarizing it into memory evidence. The prompt template is provided in the Appendix A.2. We formalize the memory retrieval process as follows:

$$M = \mathcal{R}_\theta(C, q). \tag{4}$$

where $\mathcal{R}_\theta$ denotes the memory retrieval model with parameters θ.

3.3.1 RL formulation.

The policy π_r aims to retrieve memory evidence relevant to the current query q from the memory pool. Given the dialogue context C and query q, the model produces memory evidence m_i referring to a concise factual description for query q. In this stage, we optimize the policy model π_r by maximizing the expected reward, from Equation (1), where the reward r will be detailed soon.

3.3.2 Reward Design.

We designed a dynamic reward mechanism for the memory retrieval policy π_r. Specifically, we first feed each retrieved evidence m_i from π_r into the frozen utilization policy π_u. The π_u generates an answer and simultaneously annotates whether m_i is used:

$$r_{\text{citation}}(m_i) = \begin{cases} 1, & \text{if } m_i \text{ is cited,} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

During early training, we observed that the mean citation reward rise and converge at a high level. However, as training continued, sole reliance on the citation reward led to severe reward hacking: π_r produced highly hallucinated memory evidence to deceive π_u to generate incorrect answers using them. To address this, when the citation reward converges, we introduce a semantic-matching reward between π_u's inference result and the ground-truth for π_r through LLM-as-judge:

$$r_{\text{semantic}}(q, y) = \text{LLMs}\big(q,\, y_{\text{ans}},\, y_{\text{gold}}\big). \tag{6}$$

where LLMs-Judge$(\cdot)$ produces a binary correctness score according to the evaluation template (Appendix A.2).

$$R(m_i) = \alpha\, r_{\text{citation}}(m_i) + (1-\alpha)\, r_{\text{semantic}}(q, y). \tag{7}$$

where $\alpha \in [0, 1]$ balances the importance of citation and semantic rewards, which is set to 0.6 in our experiments.

3.4 Stage2: Memory Utilization Optimization

This stage involves utilizing the memory evidence M to perform various tasks. The prompt template for this process is provided in the AppendixA.2. We formalize the memory utilization as follows.

$$y = \mathcal{U}_\theta(M, q). \tag{8}$$

where $\mathcal{U}_\theta$ denotes the memory utilization model with parameters θ.

3.4.1 RL formulation.

The utilization policy π_u aims to utilize the retrieved memory evidence M to generate reasoning and answer the current question q. Given the memory evidence Mi and the query q, the model outputs a reasoning process and a final answer y, where each m_i represents an individual piece of memory evidence. During this phase, we optimize the policy model π_u by maximizing the expected reward defined in Equation (1), with the precise formulation of r provided below.

3.4.2 Reward Design.

We employ a hybrid reward scheme that combines LLM-as-judge with the BLEU metric in this stage. First, we use GPT-4o as an LLM-based evaluator r_{LLM} to assess both the accuracy and completeness of the predicted answer y_{ans} against the gold answer y_{gold}. The evaluator returns a scalar reward:

$$\begin{aligned} r_{\text{LLM}}(q,y) &= r_{\text{corr}}(q,y) + r_{\text{compl}}(q,y). \\ r_{\text{corr}}(q,y) &= \text{LLMs}_{\text{corr}}\big(q,\, y_{\text{ans}},\, y_{\text{gold}}\big). \\ r_{\text{compl}}(q,y) &= \text{LLMs}_{\text{compl}}\big(q,\, y_{\text{ans}},\, y_{\text{gold}}\big). \end{aligned} \tag{9}$$

Although BLEU is a simple string matching metric(Papineni et al., 2002), it demonstrates high agreement with human preferences; indeed, existing studies employing BLEU alone as a reward can yield performance on par with learned reward models(Chang et al., 2025). To incorporate lexical form evaluation and reduce the indeterminacy of LLMs-based judgments, we also introduce a BLEU-based reward r_{BLEU}, calculated as the BLEU score between y_{ans} and y_{gold}:

$$r_{\text{BLEU}}(y_{\text{ans}}, y_{\text{gold}}) = \text{BLEU}\big(y_{\text{ans}},\, y_{\text{gold}}\big). \tag{10}$$

The final reward for optimizing the memory utilization policy π_u is a weighted combination of the two components:

$$R(y) = \lambda\, r_{\text{LLM}}(q,y) + (1-\lambda)\, r_{\text{BLEU}}(y_{\text{ans}}, y_{\text{gold}}) \tag{11}$$

where $\lambda \in [0,1]$ balances semantic fidelity against lexical overlap. In our experiments, $\lambda = 0.6$.

3.5 Policy Co-optimization

Figure 1 illustrates the overall training workflow, which processes mini-batches of historical context C paired with query-answer tuples (q,a).In stage 1, Freeze the utilization policy π_u. For each (C,q), sample n memory evidences $\{M_i\}_{i=1}^{n}$ via π_r. Calculate rewards of the sampled evidences by the frozen π_u using Equation (7). Update π_r with the GRPO via Equation (3) and Equation (2). In stage 2, Freeze the updated π_r. For the same (C,q), perform a single-sample retrieval ($n = 1$) to obtain M, then execute n rollouts under π_u to generate answers $\{y_i\}$. Calculate rewards via Equation (11), calculate group-normalized advantages via Equations (3), and update π_u via Equations (2)). Alternating these two stages over multiple rounds mirrors a co-evolutionary RL process (Majumdar et al., 2020; Hu et al., 2024):

Escape Retrieval Local Optima: Utilization feedback reshapes retrieval rewards, guiding π_r out of suboptimal patterns.

Avoid Reasoning Stagnation: Enhanced retrieval evidence expands π_u's exploration space, preventing narrow convergence.

Converge Toward Global Optimum: The positive feedback loop between retrieval and utilization drives both policies toward a more consistent global solution than isolated training.

The algorithm is summarized in Algorithm A.3.

4 Experiments

4.1 Datasets

We conduct our experiments on LoCoMo(Maharana et al., 2024) and LongBench (Bai et al., 2024) datasets. LoCoMo is specifically designed to evaluate long-term conversational memory capabilities and comprises ten extended multi session dialogues, each averaging approximately 600 turns and 26 000 tokens. For each dialogue, an average of 200 questions and answer pairs with ground truth annotations are provided. The questions span multiple categories, including single hop, multi hop, temporal, and open domain. LongBench is designed to evaluate long-context memory problems that demand deep understanding and reasoning. It constructed upon several widely adopted datasets, including HotpotQA(Yang et al., 2018), MuSiQue(Trivedi et al., 2022), and QASPER(Dasigi et al., 2021). It comprises six task categories, with most tasks spanning 5 000 to 15 000 tokens and totaling 4 750 test instances. In our experiments, we evaluated the SingleDoc QA and MultiDoc QA tasks.

	Single-Hop			Multi-Hop			Temporal			Open Domain		
	F1	*BLEU-1*	*J*	*F1*	*BLEU-1*	*J*	*F1*	*BLEU-1*	*J*	*F1*	*BLEU-1*	*J*
LoCoMo	25.02	19.75	–	12.04	11.16	–	40.36	29.05	–	18.41	14.77	–
MemoryBank	5.00	4.77	–	5.56	5.94	–	6.61	5.16	–	9.68	6.99	–
MemGPT	26.65	17.72	–	9.15	7.44	–	41.04	34.34	–	25.52	19.44	–
A-Mem	27.02	20.09	39.79	12.14	12.00	18.85	44.65	37.06	54.05	45.85	36.67	49.91
RAG	34.30	23.72	63.79	20.09	15.42	42.92	39.31	31.16	62.29	14.04	11.25	21.71
Mem0	38.72	27.13	67.13	28.64	21.58	51.15	47.65	38.72	72.93	48.93	40.51	55.51
MemCoRL-14b	45.06	39.48	69.61	49.52	42.73	64.02	43.55	34.40	72.54	59.83	53.64	88.35

Table 1: Comparison results on the LoCoMo dataset.

4.2 Metrics

For the LoCoMo benchmark, we employ standard F1 and BLEU-1 scores to assess lexical overlap between model outputs and ground-truth. Furthermore, we follow mem0's LLM-as-judge approach to compensate for the aforementioned metrics' lack of semantic assessment. To mitigate occasionality, we perform five independent evaluations and report the mean score. For the LongBench benchmark, we rely on its established F1 metric to evaluate the correctness of the answers.

4.3 Baselines

We categorize the baselines into two groups:

4.3.1 Memory Module Research

MemoryBank (Zhong et al., 2024) manages memory strength according to the Ebbinghaus forgetting curve, reinforcing access and decaying when unused and retrieves relevant history via a dual-tower dense retrieval model. **A-Mem** (Xu et al., 2025) constructs a dynamically interconnected memory system and uses LLM-driven methods to establish links between those notes. **MemGPT** (Packer et al., 2024) adopts an operating-system–style paradigm and issues function calls to page content between these tiers for retrieval and updates. **Mem0** (Chhikara et al., 2025) captures and manages key conversational information through coordinated extraction and update modules.

4.3.2 Full-Context Processing

LoCoMo (Maharana et al., 2024)concatenates the entire dialogue history with the query as direct input to the model, without any additional preprocessing. **Retrieval-Augmented Generation (RAG)** segments memory into fixed-length chunks (128–8192 tokens), embeds them with OpenAI's `text-embedding-small-3`, and retrieves the top k (1–2) semantically similar chunks at query time for context concatenation. **MemoryLLM** (Wang et al., 2024) compresses past context into hidden state memory tokens across all layers, forming a latent-space memory pool. **M+** (Wang et al., 2025) extends MemoryLLM by writing expired hidden states to a CPU-side long-term memory pool and incorporates a retriever that pulls relevant memories once per layer for all query heads **LLoCO** (Tan et al., 2024) employs a context encoder to compress long texts into "summary embeddings" and finetunes both encoder and model via LoRA (Low-Rank Adaptation)(Hu et al., 2022).

4.4 Training Detail

4.4.1 Base Models

We selected Qwen2.5-14B-Base and Qwen2.5-7B-Base for our experiments.

4.4.2 Training Configuration

We trained using the GRPO algorithm within the Verl framework, applying a KL factor of 1×10^{-3} and disabling entropy loss. Optimization employed AdamW with a learning rate of 1×10^{-6}, maintained constant with a linear warm-up schedule. We conducted 4 epochs of memory retrieval with a rollout batch size of 64, group size 4, and 80 training steps per co-optimization round, and 4 epochs of memory utilization with a rollout batch size of 256, group size 8, and 20 training steps per co-optimization round. Training was conducted on NVIDIA A100 80 GB GPUs, using 32 GPUs for the 14B model and 8 GPUs for the 7B model.

4.5 Main Results

The experimental results on the LoCoMo and LongBench benchmark datasets, presented in Table 1 and Table 2. In the LoCoMo benchmark, our approach, utilizing the Qwen2.5-14B-Base model,

	2wikimqa	hotpotqa	qasper	musique	Avg
MemoryLLM-7B (20k)	27.22	34.03	19.57	13.47	23.57
M+ (16k)	32.71	38.56	30.39	24.58	31.56
LLoCO	35.60	46.20	26.10	27.30	33.80
MemCoRL-7b	**54.61**	**66.90**	**42.63**	**52.15**	**54.07**

Table 2: Comparison results on LongBench dataset.

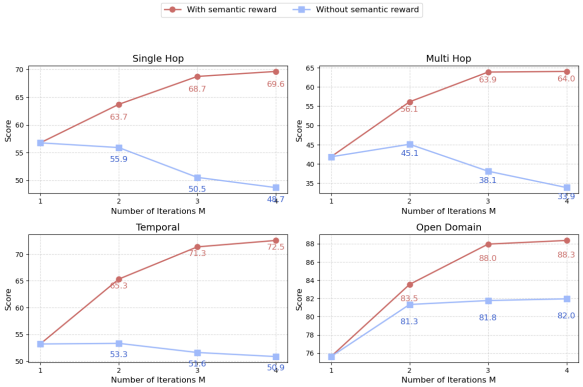

Figure 3: Ablation study on whether to include semantic rewards in memory retrieval optimization on the LoCoMo dataset using Qwen2.5-14B-Base model and LLM-as-judge metric.

outperformed the state-of-the-art baseline Mem0. Specifically, the F1 score improved by 24%, the BLEU score by 39%, and the LLM-as-judge score by 18% on average across four tasks. Similarly, in the LongBench benchmark, our MemCoRL method, employing the Qwen2.5-7B-Base model, surpassed other baselines, achieving a 63% improvement in the F1 score.

For the LoCoMo benchmark, MemoryBank exhibited the lowest performance, indicating that memory management relying solely on decay mechanisms remains insufficient. More advanced approaches, such as MemGPT, which leverages memory paging techniques, and A-Mem, which employs a dynamic memory linking mechanism, are more effective for memory establishment and management. Compared to full-context processing methods, these approaches show comparable performance to LoCoMo in single-hop and multi-hop tasks but still fall short of RAG methods. However, they outperform them in temporal and open-domain tasks. This suggests that memory module-based methods inevitably experience information loss when constructing and managing memory, while full-context processing methods incur less information loss but may face challenges related to knowledge update and integrate. In contrast, MemCoRL features less information loss, integrates evidence during the memory retrieval phase, and co-optimizes memory retrieval and utilization. Although mem0 achieved the best performance in memory module management, our method still outperformed it on average. Specifically, our method perform well on Multi-Hop tasks, where evidence is distributed and thus demands robust retrieval, and on Open-Domain tasks, which require world-knowledge reasoning and therefore robust utilization, which highlights the advanced memory retrieval and utilization capabilities of our approach. In contrast, it slightly underperforms Mem0 on the Temporal task, since we prioritized a general framework without dedicated temporal prompts or specialized training, whereas Mem0 employs stage-specific temporal cues. This results further demonstrates that, without task-specific optimizations, general-purpose models or training regimes are limited in temporal tasks (Wallat et al., 2024).

For the LongBench benchmark, which assesses context memory use with relatively brief contexts while posing more challenging queries, we compare MemCoRL with full-context processing methods, as these approaches are suited to this task. At equivalent parameter scales, our method surpassed base models and baselines leveraging latent-space memory and trained context encoders. It suggests that in context memory task, enhancing utilization is more critical than advanced management. Notably, our method delivers uniform gains across diverse document domains, underscoring its robustness to domain variation. Experimental results show that MemCoRL's superior contextual memory utilization capabilities across varied domain.

4.6 Ablation Studies

To assess each component's contribution in MemCoRL, we conduct ablation studies and analyze the results as follows:

Dynamic Reward for the Retrieval Policy Mitigate the Reward Hacking: As described above, the single citation reward will produce highly illusory retrievals. In the first co-optimization round, the citation reward nearly converged to its maximum score. Thus we compared the subsequent optimization with and without our semantic reward. As shown in the figure 3, incorporating semantic rewards further enhances performance over citation-

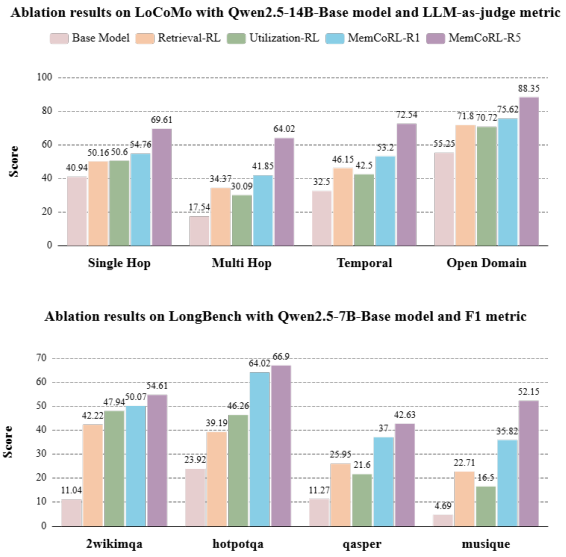

Figure 4: Ablation study on four settings: Base Model: no RL optimization; Retrieval-RL: one round of RL on the retrieval policy simply; Utilization-RL: one round of RL on the utilization policy simply; MemCoRL-R$_n$: our framework co-optimizing both policies over n rounds.

only rewards, demonstrating that dynamic rewards effectively address citation reward hacking.

Single RL is Effective for both Memory Retrieval and Memory Utilization: We conduct experiments to evaluate the performance of different settings across various tasks within two existing benchmarks, as presented in the figures 4. For the LoCoMo benchmark, optimizing only the retrieval policy resulted in an average 47% improvement over the base model in terms of LLM-as-judge performance. The improvement was particularly noticeable on more challenging tasks, such as Multi-Hop and Open Domain. Meanwhile, optimizing only the utilization policy achieved a similar improvement of 41% compared to the base model. For the LongBench benchmark, optimizing only the retrieval policy resulted in an average improvement of 110% over the base model in terms of the F1 score. Meanwhile, optimizing only the utilization policy led to an average improvement of 108% over the base model in the F1 score. The above experimental results demonstrate the effectiveness of the RL method, which is based on the specific rewards we designed for the memory retrieval and memory utilization models.

Collaborative RL Realize Co-Optimization: we track the policy model's performance across different iteration rounds for these tasks to assess the impact of alternating optimization on model

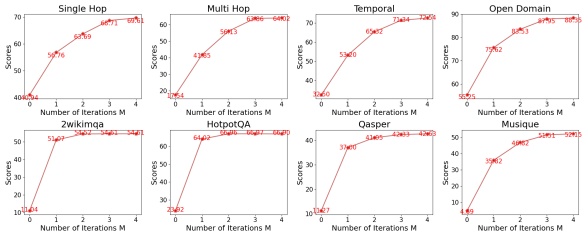

Figure 5: Ablation study on the number of collaborative RL rounds in MemCoRL, evaluated on LoCoMo with Qwen2.5-14B-Base and LLM-as-Judge metric and on LongBench with Qwen2.5-7B-Base and F1 metric.

performance. The results are presented in Figure 5, where the scores for the LoCoMo tasks represent the average of the LLM-as-judge results, and for LongBench, they correspond to the F1 score. We performed co-optimization on two policy models, resulting in an average improvement of 80% over the base model and a 20% improvement over RL alone after just one round on LLM-as-judge. Through multi-round of collaborative RL, the performance of both retrieval and utilization models continued to improve steadily toward convergence, outperforming a single round. This demonstrates the effectiveness of our proposed approach. Moreover, collaborative RL optimization, though slower to converge, provides greater performance gains for challenging tasks and larger models than for simpler tasks and smaller ones.

5 Conclusion

In this work, we address the memory management bottleneck in LLMs during long-range interactions by introducing MemCoRL, a two-stage alternating co-optimization framework based on cooperative RL. Based on extending studies that optimize retrieval using memory citation rewards with RL, we also optimize the memory utilization phase with customized reward functions. We further elucidate the bidirectional interplay between retrieval and utilization, leveraging cooperative RL to alternately co-optimize both components. We evaluate MemCoRL on diverse tasks and benchmarks, demonstrating its superiority over leading baselines. Ablation studies confirm the necessity of semantic rewards to prevent reward hacking in retrieval, the beneficial impact of RL on both retrieval and utilization, and the cumulative gains from multi-round co-optimization. Overall, MemCoRL's "retrieve–utilize" paradigm advances LLMs memory research by emphasizing effective retrieval and uti-

lization over mere storage. We anticipate that this co-optimization strategy will prove instrumental in applications such as long-horizon dialogue, knowledge tracking.

6 Limitations

MemCoRL currently optimizes memory retrieval and utilization in isolation from foundational memory storage and management mechanisms, leaving the end-to-end memory pipeline incomplete. This constraint limits its direct applicability to systems requiring integrated storage, organization, and retrieval—addressed only in future extensions via multi-agent co-optimization.

7 Acknowledgements

This work was supported by National Key R&D Program of China(Grant No. 2024YFF0907400)

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *Preprint*, arXiv:2402.14740.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

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. 2024. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Yapei Chang, Yekyung Kim, Michael Krumdick, Amir Zadeh, Chuan Li, Chris Tanner, and Mohit Iyyer. 2025. Bleuberi: Bleu is a surprisingly effective reward for instruction following. *arXiv preprint arXiv:2505.11080*.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building Production–Ready AI Agents with Scalable Long–Term Memory. *Preprint*, arXiv:2504.19413.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. *Preprint*, arXiv:2105.03011.

Muxi Diao, Lele Yang, Wuxuan Gong, Yutong Zhang, Zhonghao Yan, Yufei Han, Kongming Liang, Weiran Xu, and Zhanyu Ma. 2026. Entropy-adaptive fine-tuning: Resolving confident conflicts to mitigate forgetting. *arXiv preprint arXiv:2601.02151*.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending llm context window beyond 2 million tokens. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*.

Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. 2025. Deeprag: Thinking to retrieve step by step for large language models. *arXiv preprint arXiv:2502.01142*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Chengpeng Hu, Jialin Liu, and Xin Yao. 2024. Evolutionary reinforcement learning via cooperative coevolution. *arXiv preprint arXiv:2404.14763*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jian Hu, Jason Klein Liu, and Wei Shen. 2025. Reinforce++: An efficient rlhf algorithm with robustness to both prompt and reward models. *Preprint*, arXiv:2501.03262.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *Preprint*, arXiv:2503.09516.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. Memory OS of AI Agent. *Preprint*, arXiv:2506.06326.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning. *Preprint*, arXiv:2404.02060.

Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang, Kaitao Song, Kunlun Zhu, Yuheng Cheng, Suyuchen Wang, Xiaoqiang Wang, Yuyu Luo, Haibo Jin, Peiyan Zhang, Ollie Liu, and Jiaqi Chen. 2025. Advances

and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *Preprint*, arXiv:2504.01990.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. 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)*, pages 13851–13870, Bangkok, Thailand. Association for Computational Linguistics.

Somdeb Majumdar, Shauharda Khadka, Santiago Miret, Stephen McAleer, and Kagan Tumer. 2020. Evolutionary reinforcement learning for sample-efficient multiagent coordination. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pages 6651–6660. PMLR.

OpenAI. 2023. Gpt-4 technical report. Accessed: 2025-07-26.

Jie Ouyang, Ruiran Yan, Yucong Luo, Mingyue Cheng, Qi Liu, Zirui Liu, Shuo Yu, and Daoyu Wang. 2025. Training powerful llm agents with end-to-end reinforcement learning.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Memgpt: Towards llms as operating systems. *Preprint*, arXiv:2310.08560.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. Yarn: Efficient context window extension of large language models. *Preprint*, arXiv:2309.00071.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval–generation synergy. *arXiv preprint arXiv:2305.15294*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Ibne Farabi Shihab, Sanjeda Akter, and Anuj Sharma. 2025. Detecting and mitigating reward hacking in reinforcement learning systems: A comprehensive empirical study. *Preprint*, arXiv:2507.05619.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. 2025. Zerosearch: Incentivize the search capability of llms without searching. *Preprint*, arXiv:2505.04588.

Sijun Tan, Xiuyu Li, Shishir Patil, Ziyang Wu, Tianjun Zhang, Kurt Keutzer, Joseph E. Gonzalez, and Raluca Ada Popa. 2024. Lloco: Learning long contexts offline. *Preprint*, arXiv:2404.07979.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long T. Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. 2025. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. *Preprint*, arXiv:2503.08026.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Jonas Wallat, Adam Jatowt, and Avishek Anand. 2024. Temporal blind spots in large language models. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 683–692. ACM.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, Jingbo Shang, and Julian McAuley. 2024. Memoryllm: Towards self-updatable large language models. *Preprint*, arXiv:2402.04624.

Yu Wang, Dmitry Krotov, Yuanzhe Hu, Yifan Gao, Wangchunshu Zhou, Julian McAuley, Dan Gutfreund, Rogerio Feris, and Zexue He. 2025. M+: Extending memoryllm with scalable long-term memory. *arXiv preprint arXiv:2502.00592*.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. 2025. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning. *Preprint*, arXiv:2505.16421.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-mem: Agentic memory for llm agents. In *Advances in Neural Information Processing Systems*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025. A survey on the memory mechanism of large language model based agents. *ACM Trans. Inf. Syst.* Just Accepted.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. MemoryBank: Enhancing Large Language Models with Long–Term Memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731. AAAI Press.

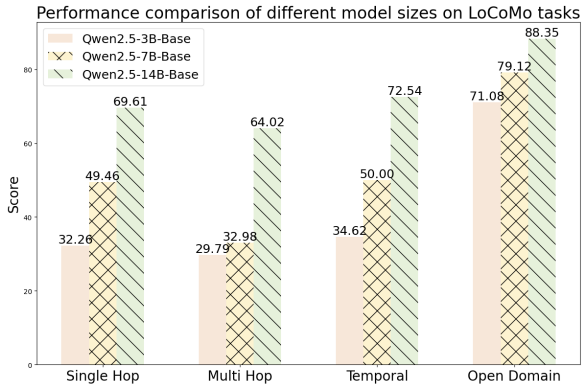

Figure 6: Ablation study on the effect of model size.

A Appendix

A.1 Additional Experiments

In addition to the main experiments presented in the paper, we conducted additional studies to assess the impact of model size on performance. Specifically, in addition to Qwen2.5-14B-Base, we use Qwen2.5-7B-Base and Qwen2.5-3B-Base as the base models and report the converged results of co-optimization for comparison. The experimental results are summarized in Figure 6. The results indicate that larger model sizes lead to greater improvements, particularly on more challenging tasks such as Multi-Hop after co-optimization.

A.2 Prompts

We provide the prompts used in MemCoRL for the three main stages of the framework: memory retrieval, memory utilization, and LLM-as-Judge evaluation. These prompts guide the model to perform different roles in a consistent and structured manner and together constitute an important part of the overall pipeline.

Figure 7 presents the memory retrieval prompt, which is used to extract memory evidence from the context.

Figure 8 presents the memory utilization prompt, which is used to answer questions based on the retrieved memory evidence.

Figure 9 presents the LLM-as-Judge prompt, which is used to provide semantic feedback during reward computation.

A.3 Algorithm

Here we present the algorithm of MemCoRL in Algorithm 1.

USER:
You are a professional assistant for long context comprehension and evidence extraction.
You can read and understand multi-turn conversations or long documents that may include images, and you can pull out precise evidence needed to answer a given question

Input:
• context as follows:{context}
• question as follows: {question}

Task:
Read the entire conversation or context, and find every passage (maybe across sessions) that is relevant to the question.
List every piece of information from the context that is related to the question.
Rewrite those supporting passages as a list of short, factual summaries that can be evidence to answer the question later.

Output:
Please place your output strictly within the following three tags:
<relative>
List all the relevant information here from the original. Note that there may be multiple related pieces from different sessions.
</relative>
<think>
Extract the key information from your relative information. Show your reasoning here.
</think>
<answer>
Give your final extracted result from relative info: a list of concise summaries of the supporting content containing the date, if it has(no extra commentary).
</answer>

Figure 7: Prompt for memory retrieval.

USER:
You are an expert in answering questions based on extracted information and relevant context.
Based on the following extracted memory evidence, please answer the question accurately and concisely.
If the memory information I provide does not contain any data that can address the current question, you must clearly point this out.

Input:
• memory evidence as follows:{evidence}
• question as follows: {question}

Output:
Your answer must follow this format:
<think>
Your reasoning: analyze whether the available memory can answer the question.
</think>
<answer>
Your direct answer to the question.
</answer>
<usage>
State clearly whether the memory supports answering the question: output True if it does, False if it does not.
</usage>
Please strictly follow the above format requirements, especially including both <usage> and </usage> tags.
True of False must in the <usage> and </usage> tags.

Figure 8: Prompt for answering queries using memory evidence.

USER:
You are an objective and impartial evaluation expert. Please assess the quality of the following question-answer pair.

Question: {question}

Model's Answer:{prediction}

Reference Answer:{reference}

Please rate the answer on a scale of 1 to 5 in the following dimensions:
1. Factual Accuracy: How consistent the answer is with the reference answer in terms of factual content.
2. Completeness: Whether the answer includes all the key information from the reference answer.
3. Overall Quality: The overall quality based on the above factors.

Your evaluation should focus on factual accuracy and content consistency between the answer and the reference answer, not on the similarity in expression style. Please provide your scores and analysis in the following format:

<scores>
Factual Accuracy: [1-5]
Completeness: [1-5]
Overall Quality: [1-5]
</scores>

Figure 9: Prompt for semantic judgment used in the dynamic reward during retrieval and in the hybrid reward during utilization.

Algorithm 1 MemCoRL

Require: History context C; question-answer pairs $D = \{(q_1, a_1), (q_2, a_2), \ldots, (q_N, a_N)\}$; memory retrieval policy π_r; memory utilization policy π_u; number of iterations M; number of evidence rollouts m; number of answer rollouts n; learning rate η; KL coefficient β.

Initialize memory retrieval policy π_r and memory utilization policy π_u.
for $t = 1$ to M **do**
  Collect evidence rollout samples:
  for each question-answer pair $(q, a) \in D$ **do**
    Generate m evidence samples $\{m_j\}_{j=1}^{m}$ using $\pi_r(q, C)$.
  end for
  Compute rewards for evidence samples:
  for each evidence m_j **do**
    $R(m_j) = \alpha r_{\max}(m_j) + (1 - \alpha) r_{\max}(q, y)$.
  end for
  Optimize the retrieval policy π_r:
  Compute advantages $A_{m_j} = \text{normalize}(R_{m_j})$.
  Update π_r with Eq. (2).
  Collect evidence rollout samples for utilization:
  for each question-answer pair $(q, a) \in D$ **do**
    Generate evidence m_i using $\pi_r(q, C)$.
  end for
  Collect answer rollout samples:
  for each question-answer pair $(q, a) \in D$ **do**
    Generate n answers $\{a_i\}_{i=1}^{n}$ using $\pi_u(q, m_i)$.
  end for
  Compute rewards for answer samples:
  for each answer a_i **do**
    $R(y) = \lambda r_{\text{LLM}}(q, y) + (1 - \lambda) r_{\text{BLEU}}(y, y_{\text{gold}})$.
  end for
  Optimize the utilization policy π_u:
  Compute advantages $A_{a_i} = \text{normalize}(R_{a_i})$.
  Update π_u with Eq. (2).
end for
Output: optimized policies π_r and π_u.