Learning How and What to Memorize: Cognition-Inspired Two-Stage Optimization for Evolving Memory

Derong Xu[1,2], Shuochen Liu[1], Pengfei Luo[1], Pengyue Jia[2], Yingyi Zhang[2,3], Yi Wen[2], Yimin Deng[2,4], Wenlin Zhang[2], Enhong Chen[1], Xiangyu Zhao[2,*], Tong Xu[1,*]
[1]University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence, [2]City University of Hong Kong
[3]Dalian University of Technology, [4]Xi'an Jiaotong University
derongxu@mail.ustc.edu.cn, xianzhao@cityu.edu.hk, tongxu@ustc.edu.cn

Abstract

Large language model (LLM) agents require long-term user memory for consistent personalization, but limited context windows hinder tracking evolving preferences over long interactions. Existing memory systems mainly rely on static, hand-crafted update rules; although reinforcement learning (RL)-based agents learn memory updates, sparse outcome rewards provide weak supervision, resulting in unstable long-horizon optimization. Drawing on memory schema theory and the functional division between *prefrontal regions* and *hippocampus regions*, we introduce MemCoE, a cognition-inspired two-stage optimization framework that learns **how** memory should be organized and **what** information to update. In the first stage, we propose **Memory Guideline Induction** to optimize a global guideline via contrastive feedback interpreted as textual gradients; in the second stage, **Guideline-Aligned Memory Policy Optimization** uses the induced guideline to define structured process rewards and performs multi-turn RL to learn a guideline-following memory evolution policy. We evaluate on three personalization memory benchmarks, covering explicit/implicit preference and different sizes and noise, and observe consistent improvements over strong baselines with favorable **robustness**, **transferability**, and **efficiency**[1].

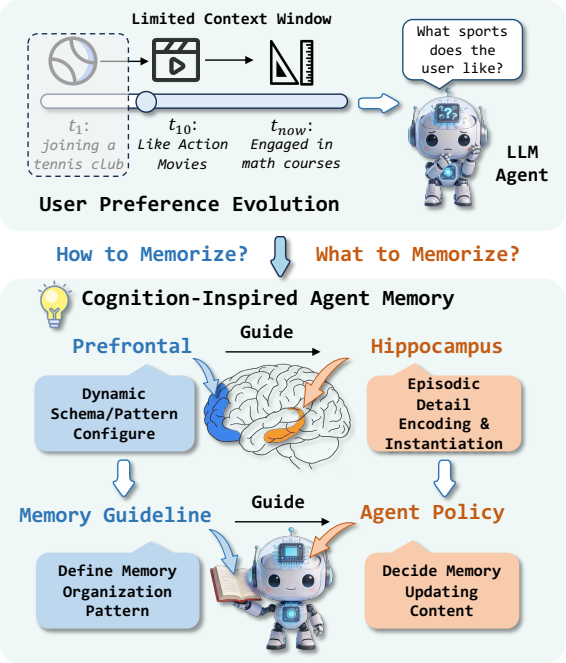


Figure 1: Top: With a limited context window, the agent fails to capture preferences. Bottom: Inspired by `Prefrontal` → `Hippocampus`, we decouple agent memory into `Memory Guideline` (for organizing) → `Agent Memory` (for updating).

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities as conversational agents in various real-world applications, such as personal assistant (Li et al., 2024b), customer service (Tan et al., 2025b; Zhang et al., 2025a), and education (Wen et al., 2024). In these settings, adaptive personalized interaction depends on the agent's ability to continuously integrate information about a user's evolving preferences and habits (Jiang et al., 2025a,b). However, the context window prevents LLMs from retaining and exploiting the full history of dialogue (Yen et al., 2024), and simply storing and retrieving past dialogue snippets struggles to capture such dynamic patterns (Li et al., 2025b). This limitation highlights the necessity of maintaining an external memory that preserves important information over time, enabling consistent and personalized responses (Chhikara et al., 2025; Zhong et al., 2024; Deng et al., 2026).

Many existing LLM agent memory systems typically build a workflow that converts raw dialogue into external memory banks (Xu et al., 2026, 2025b; Fang et al., 2025). Yet these methods rely

*Corresponding authors.
[1]https://github.com/Applied-Machine-Learning-Lab/ACL2026_MemCoE

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 44987–45011
July 2-7, 2026

on static and predefined pipelines for extraction rules, making it difficult to learn from interaction feedback or adapt to user behavior. To solve this, other works treat memory operations as learnable actions and train an end-to-end memory update policy with reinforcement learning (RL) (Yu et al., 2025; Yan et al., 2025; Wang et al., 2025b). While more adaptive, memory updates typically involve free-form edits over what to write or forget. When guided by only simple instructions and optimized with sparse and delayed outcome-level rewards, the policy is weakly constrained and faces a large action space, making exploration and long-horizon optimization challenging. This often results in unstable training and increased data requirements (Zhang et al., 2025d), motivating the need for more effective mechanisms for memory organization and updating.

Memory schema theory (Alba and Hasher, 1983) in cognitive psychology, as shown in Figure 1 bottom, offers a perspective for understanding how human memory is organized and updated. Specifically, the theory suggests a functional division of labor between brain systems: **Prefrontal regions** dynamically select and configure an appropriate schema based on the current context, thereby shaping expectations and attentional priorities, while the **Hippocampus regions** (Teyler and DiScenna, 1986) instantiate this schema by encoding the concrete episodic details of ongoing experience. Importantly, this division is advantageous because it maintains a **stable schema-level organizing prior** that guides attention and structuring, while allowing the hippocampus system to flexibly encode context-specific episodic details within that scaffold. From this perspective, the separation naturally decouples how memory is controlled (i.e., the organization patterns) and what is stored (i.e., the update content). Motivated by this mechanism, we ask the following question:

> 💡 *Can we build a schema-based agent memory system whose organization and updating mechanisms evolve in a manner analogous to human brain?*

To answer this question, in this paper, we introduce **MemCoE**, a two-stage optimization framework inspired by a functional analogy to human memory, enabling the agent to learn **how** memory should be organized, and **what** content should be stored and updated, by optimizing a memory guideline and a policy for evolving memory. Our approach maintains a user memory bank that evolves alongside the dialogue. In the first stage, to simulate the Prefrontal regions, we propose ♣ **Memory Guideline Induction (MGI)**, which treats the instruction prompt as a global natural-language parameter and optimizes it via two key techniques: (i) interpreting contrastive feedback over memory-augmented trajectories as textual gradients and (ii) aggregating these gradients at the batch level, thereby inducing a domain-agnostic textual guideline. In the second stage, we propose ♠ **Guideline-Aligned Memory Policy Optimization (GMPO)**, which encodes instance-specific episodic details via guideline-aligned policy optimization. GMPO utilizes the optimized guideline to define guideline-aligned rewards and performs multi-turn RL over memory-augmented trajectories to train the memory-evolution policy end-to-end, thereby jointly encouraging adherence to the guideline and determining what content should be memorized. Crucially, the first stage induces a guideline that defines a stable set of memory operations, effectively constraining the action space explored by the policy. Given this constrained space, the second stage can focus on process-level guideline reward, learning to invoke the guideline-specified operations with appropriate content.

We empirically evaluate **MemCoE** on three personalization memory benchmarks (PersonaMem, PrefEval, and PersonaBench), spanning explicit and implicit preference and increasingly noisy evidence sources, where accurate answering requires tracking evolving user states over extended histories. Across these settings, **MemCoE** consistently outperforms strong baselines built on static memory templates or RL-based memory updating, while remaining **efficient** in memory evolution and **scalable** to longer contexts and more rounds. Moreover, the induced guideline exhibits strong **transferability** across LLMs, supporting **robust generalization** under distribution shifts in query type.

2 Related Work

Memory for LLM Agents. Memory has become a foundational capability for LLM agents, supporting long-horizon understanding, continual adaptation in complex environments(Zhang et al., 2024d; Wu et al., 2025; Hu et al., 2025). To equip LLM agents with memory, most methods construct an explicit memory bank supported by primitives for

segmentation (Pan et al., 2025), summarization (Kim et al., 2024; Wang et al., 2025a; Team, 2023; Liu et al., 2023; Lu et al., 2023; Rasmussen et al., 2025), compression (Chen et al., 2025a; Lee et al., 2024; Xu et al., 2023; Chen et al., 2025b), and forgetting/updating to maintain long-term quality (Zhong et al., 2024; Li et al., 2024a). To improve retrieval quality, several approaches build structured memory indices such as trees (Rezazadeh et al., 2024; Sarthi et al., 2024) and graphs (Gutiérrez et al., 2025; Chhikara et al., 2025; Wang and Chen, 2025; Xu et al., 2024). Beyond generic storage, personalization-oriented methods emphasize capturing user profiles and preferences for downstream conditioning (Zhang et al., 2026a; Xu et al., 2025a; Zhang et al., 2025e; Du et al., 2024; Fang et al., 2025; Qian et al., 2025). Another thread focuses on experience memory, where agents memorize successful or failed trajectories to improve future decision-making (Packer et al., 2023; Wang et al., 2025c; Tang et al., 2025a; Ouyang et al., 2025; Zhao et al., 2024; Zhang et al., 2025b; Gao et al., 2025; Jia et al., 2024). Despite strong performance, these methods rely on hand-crafted heuristics, which can be brittle under non-stationary user behaviors. Our method is orthogonal to memory construction and storage, and is broadly compatible with existing memory backends as a general mechanism for memory evolution to support long-term personalization.

RL for Memory. Recent works treat memory operations as a sequential decision problem and optimize it with reinforcement learning (Zhang et al., 2026b; Wang et al., 2025b; Zhou et al., 2025; Yan et al., 2025; Long et al., 2025; Yuan et al., 2025; Zhang et al., 2025f; Liu et al., 2025; Li et al., 2025a). For example, RMM (Tan et al., 2025b) learns to manage long-term personalized memory via reflective update and retrieval; MemAgent (Yu et al., 2025) uses RL to learn a memory agent that maintains a fixed-length context by selectively preserving/overwriting long dialogue history; MEM1 (Zhou et al., 2025) trains memory and reasoning synergy to form compact memory for efficient long-horizon agent; MemGen (Zhang et al., 2025c) proposes generative latent memory that weaves experience into reusable memory tokens for self-evolving agents. However, these approaches typically rely on final success/answer as sparse rewards, and lack process-level rewards that directly guide how memory should be updated. We address this by introducing *guideline-aligned rewards*, which provide structured learning signals for memory evolution.

Prompt Optimization. A growing line of work treats prompts as optimizable natural-language parameters, iteratively refining them via model-generated feedback rather than numerical gradients (Yuksekgonul et al., 2025; Yang et al., 2023; Pryzant et al., 2023; Shinn et al., 2023; Tang et al., 2025b; Zhang et al., 2024c,b,a). The common pattern involves evaluating the current prompt, generating natural-language edit signals (textual gradients), and applying them to produce improved variants. For instance, TextGrad (Yuksekgonul et al., 2025) uses LLM-generated textual gradients for iterative prompt refinement; OPRO (Yang et al., 2023) treats the LLM as a black-box optimizer that proposes and scores prompt candidates; Reflexion (Shinn et al., 2023) converts past failures into self-reflection feedback carried forward to guide future actions. While related in spirit, MGI differs in that it optimizes a global memory-evolution guideline over long histories rather than a single-turn prompt, stabilizes updates via contrastive diagnosis and batch aggregation, and interfaces with Stage-2 RL through guideline-aligned process rewards to jointly optimize memory update policies and reinforced behaviors.

3 Preliminary

We consider a conversational setting in which a user interacts with an assistant agent over time. Let h_t denote the t-th dialogue snippet, and let the cumulative interaction history be represented as a dialog set $\mathcal{H} = \{h_1, h_2, \ldots, h_t\}$. To support long-term personalization, the system maintains a **user memory bank** $\mathcal{M}_t$, a textual representation that evolves dynamically as new user behaviors and preferences emerge. Beyond dialogue content h_t, we incorporate a learnable *memory-update prompt* $\mathcal{S}$ as a parameterized system component that regulates how memory is updated. Formally, the overall memory bank-evolution process is summarized in generic form with an evolution module $\mathcal{T}$:

$$\mathcal{M}_{t+1} = \mathcal{T}(\mathcal{M}_t, h_t; \mathcal{S}, \phi), \tag{1}$$

where ϕ denotes the parameters of LLM. The evolution operator $\mathcal{T}$ encapsulates the mechanisms for incorporating new information, refining existing entries, and removing outdated or inconsistent content. This formulation treats user memory as a

continuously adapting latent structure aligned with the user's evolving profile.

Given a task or query input x, the agent $\mathcal{A}$ generates a personalized response conditioned on x and the current memory state $\mathcal{M}_t$: $y_t = \mathcal{A}(x, \mathcal{M}_t)$, indicating that $\mathcal{M}_t$ serves as auxiliary context modulating the agent's behavior. The central challenge in our task, therefore, lies in designing a principled mechanism $\mathcal{T}$ that allows $\mathcal{M}_t$ to evolve coherently with $\mathcal{H}$, enabling the agent to maintain stable, accurate, and temporally consistent user representations throughout long-term interaction.

4 Methodology

We propose a two-stage framework for learning an effective memory-evolution mechanism. Instead of hand-crafting the update rule inside the evolution operator $\mathcal{T}$, we treat the memory update instruction as an optimizable natural-language parameter and learn it from data. In the first stage, **Memory Guideline Induction**, we learn how the agent performs memory operations by inducing a high-quality textual guideline. Subsequently, we further optimize what to store in accordance with this guideline. The two stages are shown in Figure 2.

4.1 Memory Guideline Induction

Existing implementations of memory-evolution methods typically rely on manually designed templates or prompts that prescribe how new dialogue segments should modify the user's memory. Such heuristic guidelines are brittle and lack the ability to adapt across domains, user styles, and annotation conventions. Inspired by schema-based human memory mechanisms, we instead treat the instruction prompt $\mathcal{S}$ as a global natural-language parameter encoding a structured policy for memory operations, and aim to learn it from data. Consequently, the objective of the Memory Guideline Induction stage is therefore to induce an optimized guideline $\mathcal{S}^\star$ that teaches the agent how to perform memory evolution correctly.

Contrastive feedback as textual gradient. Firstly, we use a training set where each example provides a dialogue history $\mathcal{H}$ and a query x. At optimization step k, given the current guideline $\mathcal{S}^{(k)}$, we first run the memory-evolution operator over the history and then perform multiple forward propagations of the agent to answer the query. This produces a set of trajectories $\{\tau_i\}$, where each trajectory τ_i contains the query x, the intermediate memory states, and a candidate response y_i. Using task supervision or environment feedback, we select at least one correct trajectory τ^+ and treat the remaining, partially plausible but suboptimal ones as contrastive negatives $\{\tau_j^-\}$. To obtain contrastive feedback, we apply a predefined feedback instruction $\mathcal{P}_g$ that compares the correct trajectory τ^+ with the negative trajectories $\{\tau_j^-\}$, highlighting the desired properties of τ^+ and the typical errors in the negatives. The resulting natural-language contrastive reflection serves as a **textual gradient**, guiding the iterative refinement of the guidelines:

$$g^{(k)} = \mathrm{Grad}\big(\tau^+, \{\tau_j^-\}; \mathcal{P}_g\big). \tag{2}$$

This textual gradient is then used to update $\mathcal{S}^{(k)}$, guiding the agent toward more reliable and task-aligned trajectories.

Batch-level gradient aggregation. To obtain a stable and general update signal, we aggregate textual gradients across a mini-batch B of training examples. Each $g^{(k)}$ provides a localized critique about how $\mathcal{S}^{(k)}$ should change for a specific $(\mathcal{H}, x)$; the $\mathrm{Aggr}(\cdot)$ operator synthesizes these instance-level signals into a single, abstract update direction:

$$G^{(k)} = \mathrm{Aggr}\big(\{g^{(k)}\}_{(\mathcal{H},x)}; \mathcal{P}_a\big), \tag{3}$$

where $\mathrm{Aggr}(\cdot)$ can be instantiated as a summarization and abstraction procedure, guided by an aggregation prompt $\mathcal{P}_a$, that identifies common failure patterns and consolidates them into a guideline-level modification proposal.

Optimization objective. By applying the merged textual gradient $G^{(k)}$, the guideline $\mathcal{S}^{(k)}$ is refined through an optimization operator that performs natural-language editing.

$$\mathcal{S}^{(k+1)} = \mathrm{Optim}\big(\mathcal{S}^{(k)}, G^{(k)}; \mathcal{P}_o\big). \tag{4}$$

Conceptually, this iterative procedure performs gradient-like steps on an underlying contrastive objective that promotes answers aligned with the positive references and penalizes confusing them with the negatives. Let $\mathcal{R}(\cdot)$ denote a reward function; in our implementation, it simply indicates whether the output is correct. Under this view, the induced guideline $\mathcal{S}^\star$ can be regarded as an approximate maximizer of the expected reward:

$$\mathcal{S}^\star = \arg\max_{\mathcal{S}} \mathbb{E}_{(\mathcal{H},x)}\Big[\mathcal{R}\big(\tau^+, \{\tau_j^-\}; \mathcal{S}\big)\Big]. \tag{5}$$

This yields a memory guideline that encodes effective principles for guiding downstream memory evolution, which are provided in Figure 19.

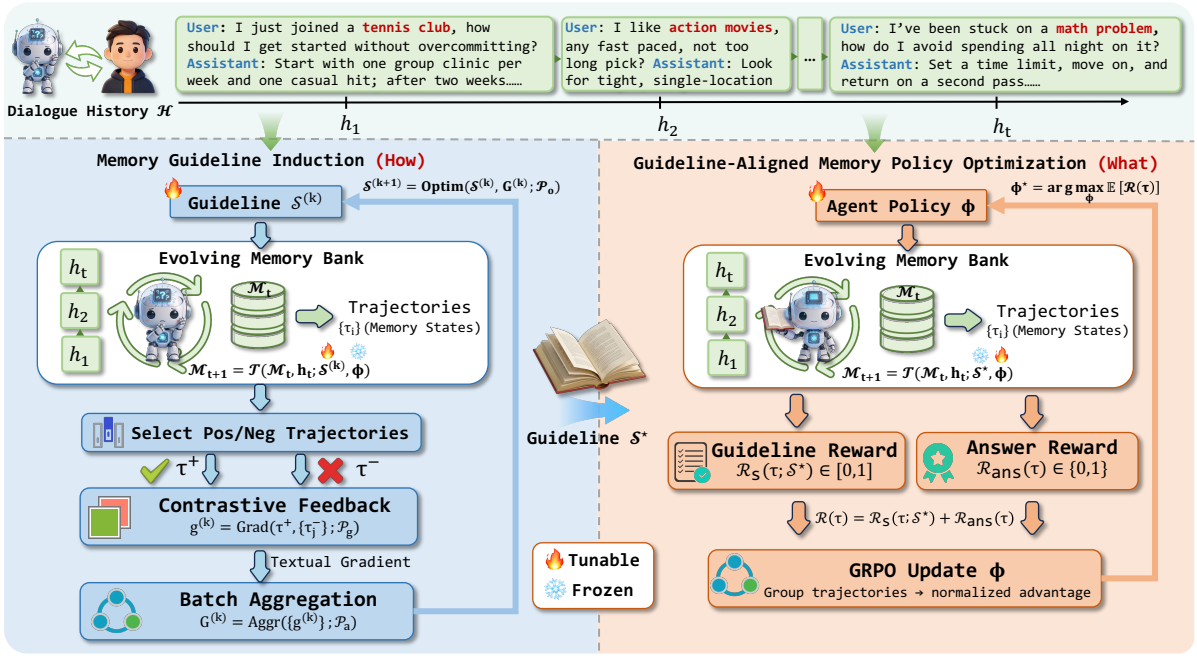


Figure 2: Overview of our proposed **MemCoE**. It performs two-stage optimization for evolving user memory: (1) Memory Guideline Induction (**MGI**) iteratively refines a natural-language guideline; (2) Guideline-Aligned Memory Policy Optimization (**GMPO**) fixes the induced guideline to define guideline-aligned rewards and applies multi-turn GRPO to learn what information to update in evolving memory bank.

4.2 Guideline-Aligned Memory Policy Optimization

Building on the induced guideline $\mathcal{S}^\star$, the second stage focuses on optimizing *what* to store in the user memory. We fix $\mathcal{S}^\star$ and regard the parameters ϕ of the evolution operator $\mathcal{T}$ and the agent $\mathcal{A}$ as a unified policy over memory-augmented trajectories. For each training instance $(\mathcal{H}, x)$, rolling out the system under $\mathcal{S}^\star$ produces a trajectory τ that interleaves memory updates $\mathcal{M}_{t+1} = \mathcal{T}(\mathcal{M}_t, h_t; \mathcal{S}^\star, \phi)$ and intermediate responses $y_t = \mathcal{A}(x, \mathcal{M}_t)$, culminating in a final answer used for evaluation.

Guideline-aligned rewards. Our first signal is a *Guideline-aware* reward that explicitly enforces the guideline induced by $\mathcal{S}^\star$. For each memory-update segment in τ, we parse the model output and prompt LLM to score whether the update strictly follows the prescribed output format (e.g., required fields, tags, and structure). These signals are aggregated into a dense guideline reward $\mathcal{R}_S(\tau; \mathcal{S}^\star) \in [0, 1]$, which encourages $\mathcal{T}$ to produce guideline-aligned, well-structured memory edits rather than arbitrary free-form text. Second, an answer reward $\mathcal{R}_{\text{ans}}(\tau) \in \{0, 1\}$ measures task correctness by directly comparing the final response in τ with the reference answer (e.g., exact or judged match), yielding a simple correctness signal used to align the memory policy with downstream performance. The overall trajectory reward combines the two components as $\mathcal{R}(\tau) = (1-\lambda) * \mathcal{R}_S(\tau; \mathcal{S}^\star) + \lambda * \mathcal{R}_{\text{ans}}(\tau)$, where λ balances guideline fidelity and answer accuracy.

Policy optimization. We optimize ϕ using Group Relative Policy Optimization (GRPO) over groups of trajectories on multi-conversation memory evolution. For each $(\mathcal{H}, x)$, GRPO samples a group of trajectories, computes group-normalized advantages from $\mathcal{R}(\tau)$, and applies a clipped policy-gradient update. Abstractly, the learned guideline-aligned memory policy is obtained by

$$\phi^\star = \arg\max_\phi \mathbb{E}_{(\mathcal{H},x)\sim\mathcal{D},\ \tau\sim\pi_\phi(\cdot|\mathcal{H},x;\mathcal{S}^\star)} \big[\mathcal{R}(\tau)\big]. \tag{6}$$

More details can be seen in Appendix A. In this way, the second stage learns a memory-evolution policy that follows the induced guideline while selectively storing information that is most beneficial for downstream interaction quality.

5 Experiments

5.1 Experimental Settings

Datasets and Metrics. We evaluate on three personalization memory benchmarks: Person-

Method	PersonaMem		PrefEval		PersonaBench (Noise Level)				Overall
	32K	128K	Explicit	Implicit	w/o Noise	0.3	0.5	0.7	
Long Context	34.36	25.05	31.70	30.80	29.00	19.10	17.83	13.00	26.90
RAG	48.67	38.90	47.80	32.40	29.09	28.16	24.31	23.00	36.68
Mem0	48.53	39.67	57.60	46.40	17.60	19.75	19.22	17.80	38.23
A-Mem	48.26	38.22	62.30	52.80	30.32	28.56	25.19	24.45	42.64
LightMem	50.72	39.93	64.20	54.80	19.08	18.74	19.65	17.80	41.21
MemAgent	53.58	43.59	72.30	63.60	20.05	19.36	16.51	17.92	45.00
Mem-α	53.37	42.86	71.90	62.50	19.92	17.02	16.43	15.59	44.19
MemCoE (Ours)	**57.06**	**47.24**	**81.30**	**69.90**	**32.27**	**29.89**	**25.99**	**25.09**	**52.02**

Table 1: **Overall comparison across eight evaluation settings.** We report results on PersonaMem (32K/128K) (In-Domain), PrefEval (Explicit/Implicit) (Out-of-Domain), and PersonaBench under different noise levels (Out-of-Domain). Higher is better. The best results are highlighted in **bold**.

aMem (Jiang et al., 2025a), PrefEval (Zhao et al., 2025), and PersonaBench (Tan et al., 2025a). PersonaMem measures preference evolution over long multi-session histories at different context scales. PrefEval emphasizes explicit vs. implicit preference multi-choice queries (1,000 each) with 50 inserted turns. PersonaBench tests personalized retrieval and QA over heterogeneous, noisy user corpora. We report accuracy on PersonaMem and PrefEval, and F1 on PersonaBench. Details are reported in Appendix C.

Baselines. We compare our approach against a diverse set of baselines. **LongContext** directly feeds as much of the raw interaction history. **RAG** denotes a retrieval-augmented generation setup that indexes all historical dialogue snippets in a vector store and retrieves the top-K relevant segments. To study external memory architectures, we further include three retrieval-based memory methods, **Mem0**, **A-Mem**, and **LightMem**, which maintain an external memory bank and update it. We also compare with two reinforcement-learning-based memory agents, **MemAgent** and **MEM-α**, which explicitly learn memory evolution actions. All baselines are implemented on top of the same backbone and evaluated under the same data split and hyperparameter setup for a fair comparison.

Implementation Details. We mainly leverage `Qwen2.5-7B-Instruct` (Yang et al., 2024) as the backbone LLM for all methods (MEM-α uses `Qwen3-4B`). For retrieval, we adopt `all-MiniLM-L6-v2` (Wang et al., 2020) and retrieve the Top-10 candidates. We construct training data by sampling 300 examples from PersonaMem. During training, we use retrieved dialogues as context to reduce computation when learning memory evolution; during inference, we feed the full dialogue history as context. Each memory-evolving round inputs a 4K-token chunk. All baselines are implemented using their publicly available codebases. For a fair comparison, MemAgent and MEM-α use their publicly released checkpoints, additionally training the same 300 PersonaMem training samples used by our method. All experiments are conducted on four A6000 GPUs. The same hyperparameters are shared across all retrieval-based methods. Hyperparameters are reported in Appendix B.

Setting	PersonaMem		PrefEval	
	32K	128K	Explicit	Implicit
MemCoE	**57.06**	**47.24**	**81.30**	**69.90**
w/o CF	56.44	46.33	78.30	68.10
w/o GR	56.24	46.06	79.50	68.30
w/o MGI	54.81	44.50	73.20	63.60
w/o GMPO	53.37	43.97	77.40	66.20
w/o ALL	48.47	39.09	71.70	60.60

Table 2: **Ablation Study. CF**: a contrastive feedback for textual-gradient guideline induction. **GR**: a guideline reward for enforcing the induced schema during memory updates.

5.2 Overall Evaluation

Overall Comparison with Baselines. Table 1 shows that **MemCoE** achieves the best overall score across the three benchmarks, indicating that learning a memory-evolution mechanism is more effective than fixed context inclusion or manually designed update heuristics. Specifically, Long Context degrades notably under noisy histories, while **MemCoE** captures user preference by filtering irrel-

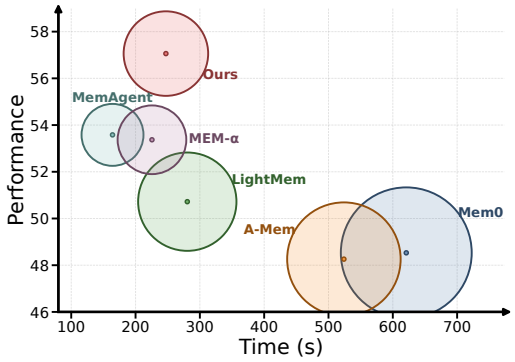


Figure 3: Efficiency analysis on PersonaMem. We report the performance–time balance of memory construction/evolution over 20 dialogue histories (32K), where circle size indicates the standard deviation of runtime.

Method	Qwen2.5-7B Instruct	gpt-4o -mini	gemini-2.5 -flash	GPT-5
RAG	48.67	47.44	61.15	63.80
A-Mem	48.26	48.47	62.37	64.42
▼ *Optimized w/ Qwen2.5-7B-Instruct*				
`MemCoE`	**53.37**	52.56	64.62	66.67
▼ *Optimized w/ gpt-4o-mini*				
`MemCoE`	52.56	**54.19**	**64.83**	**67.28**

Table 3: Cross-LLM transferability of `MGI` optimized guidelines (without RL). We optimize the guideline with one LLM and evaluate with different LLMs.

evant content when evolving memory. Compared with explicit memory-bank baselines (Mem0, A-Mem, LightMem), `MemCoE` delivers larger improvements on both PersonaMem and PrefEval. RL-based memory agents (MemAgent, Mem-α) are competitive, yet they still lag behind in overall performance. This trend aligns well with our two-stage design: `MGI` induces a transferable guideline for memory evolution, while `GMPO` learns to retain preference-relevant information under the guideline, which demonstrates the effectiveness of our method for stable long-horizon personalization.

Generalizations Across Settings. Across the eight settings in Table 1, `MemCoE` shows strong generality, consistently outperforming baselines on both in-domain tasks (PersonaMem, 32K→128K) and out-of-domain tasks (PrefEval Explicit/Implicit; PersonaBench with increasing noise) evaluations, consistent with our two-stage design: `MGI` learns stable memory organizations, while `GMPO` retains preference-relevant information. We reported results in different categories in Appendix D.

5.3 Ablation Study

To further investigate the impact of each designed module, we conduct ablation study on two prevalent datasets. As depicted in Table 2, it shows that the full model performs best on both long-context PersonaMem and distractor-heavy PrefEval, indicating that our two-stage framework is necessary for stable memory evolution. Removing **CF** or **GR** causes consistent but smaller drops (e.g., PersonaMem 32k: 57.06→56.44 / 56.24; PrefEval Explicit: 81.30→78.30/79.50), suggesting that contrastive textual feedback and guideline-aligned rewards both improve update reliability. In contrast, ablating either stage yields more significant degradations: removing `MGI` most strongly hurts preference retention and inference performance (PrefEval Explicit/Implicit: 81.30/69.90→73.20/63.60), while removing `GMPO` more severely impacts long-horizon tracking on PersonaMem (32k/128k: 57.06/47.24→53.37/43.97). Finally, **w/o ALL** collapses performance across benchmarks (PersonaMem 32k: 48.47; PrefEval Explicit: 71.70), confirming that learned guidelines plus guideline-aligned policy optimization are both critical.

5.4 Efficiency Analysis

Considering the use of LLMs, we further explore the efficiency of our proposed method. Figure 3 illustrates the performance-time trade-off of memory construction/evolution. Our `MemCoE` achieves the best performance while remaining among the faster approaches, indicating a favorable efficiency frontier rather than a pure accuracy-at-any-cost gain. This advantage is consistent with the design of `MemCoE`: instead of repeatedly invoking an LLM to separately extract and merge memory entries (e.g., A-Mem and Mem0), our approach **internalizes extraction, update, and forgetting behaviors into the model's memory evolution process**, reducing integration overhead. In contrast, MemAgent and MEM-α run quickly but fall behind in performance, suggesting that their memory update mechanism cannot reliably maintain useful user information.

5.5 Cross-LLM Transferability of Guidelines

We also investigate the transferability of different LLMs in our method, particularly focusing on the LLMs used for optimization and evaluation. As shown in Table 3, we select various mainstream LLMs to evaluate whether **optimized guidelines** transfer across different backbone LLMs. The re-

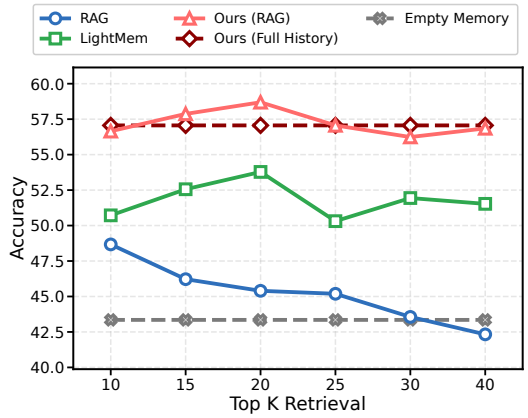


Figure 4: Retrieval Top-K on PersonaMem (32K).

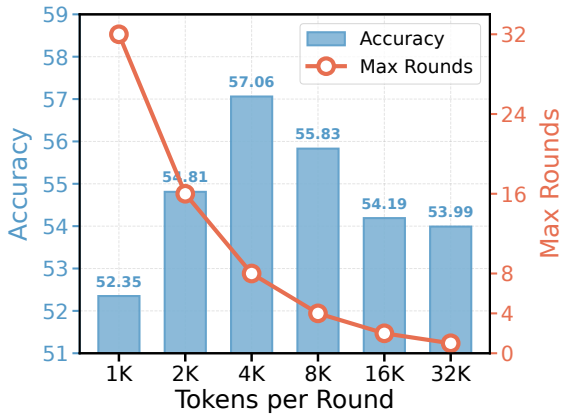


Figure 5: Effect of tokens per evolve round on PersonaMem (32K).

sults show that, across all four LLMs, both **MGI** variants consistently outperform the baselines (RAG and A-Mem), indicating that the learned guideline captures model-agnostic memory-update principles rather than overfitting to a specific LLM. Notably, optimizing with `gpt-4o-mini` generalizes strongly and achieves the best numbers on three backbones, including `GPT-5` and `gemini-2.5-flash`. Overall, these results support that **MGI** produces a guideline that is portable across LLMs, making it practical to optimize once and deploy under different backbone choices.

5.6 Comparison on Different Retrieval

In Figure 4, we examine how Top-K retrieval affects different methods on the PersonaMem dataset. Across all Top-K, both inference modes of our approach (performing memory evolution on retrieved context, **Ours (RAG)**, or on full history, **Ours (Full History)**) remain consistently strong and clearly outperform the baselines. Notably, **Ours (RAG)** peaks around K=20 and even surpasses the full-history variant, which suggests that retrieval can be beneficial when it filters irrelevant context and reduces noise for better memory evolution. In contrast, vanilla RAG degrades as K increases and even drop below Empty Memory, indicating that simply adding more retrieved content may introduce distractors that hurt downstream decisions. Overall, these results show that retrieval is not sufficient on its own; it achieves its best effect when coupled with **MemCoE** to transform retrieved evidence into coherent memory.

5.7 Impact of Per-Round Token Budget on Memory Evolution

Since the per-round token budget directly determines inference cost in real-world deployment, we study its impact on memory evolution in Figure 5. When token budget is too small (e.g., 1K–2K), system must split history into many rounds, and repeated evolve operations can accumulate errors and trigger uncontrolled forgetting, which ultimately hurts accuracy. In contrast, increasing budget initially tends to improve performance by reducing the required number of rounds and stabilizing the update dynamics. However, excessively large budgets (8K–32K) can make each evolution step more challenging, as the resulting context becomes more complex, thereby increasing the difficulty of processing information within a single pass. Overall, the results suggest a clear trade-off: effective memory evolution requires a moderate per-round token budget that avoids both excessive update frequency and overly complex single-step contexts.

5.8 Effect of Guideline Quality

Finally, to gain a more intuitive understanding of the impact of guideline quality, we compare guidelines of different quality levels. As shown in Figure 6, improving the quality of the memory-update guideline consistently strengthens downstream performance. Starting from a manually written prompt, an LLM rewrite yields a moderate gain, suggesting that surface-level prompt refinement helps but remains limited. Note that, the guideline induced by **MGI** achieves the best results on both settings, reaching 53.28 on 32K and 43.76 on 128K, which corresponds to relative improvements of +10.4% and +11.3% over the manual prompt, respectively. The error bars across three seeds indicate that these gains are stable rather than driven by randomness.

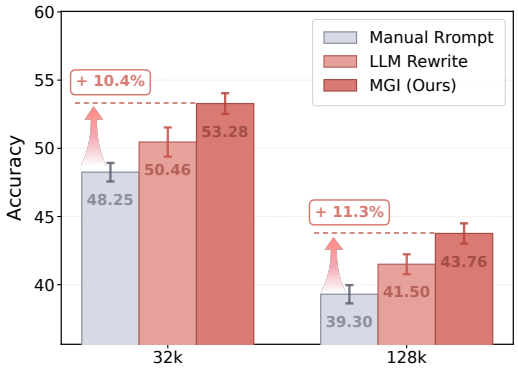


Figure 6: Impact of prompt quality on PersonaMem, averaged over three random runs with different seeds; error bars indicate standard deviation.

6 Conclusion

Inspired by memory schema theory that highlights *prefrontal regions* and *hippocampus regions*, we present **MemCoE**, a two-stage optimization framework that decouples **how to organize memory** from **what to store**. Specifically, **MemCoE** first induces a transferable, schema-consistent guideline for memory evolution, and then optimizes a guideline-aligned memory policy to decide what to retain, update, or forget across multi-session interactions. Extensive experiments on three personalization memory benchmarks show that **MemCoE** consistently outperforms strong retrieval-based memory bank, and RL-based memory-agent baselines, while remaining robust under longer histories and noisier evidence. Overall, the results support that coupling an explicit evolution guideline with policy optimization yields a practical improvement in **efficiency**, **robustness**, and **transferability** for evolving user memory in conversational agents.

Acknowledgements

This work was supported in part by the grants from National Science and Technology Major Project (No. 2023ZD0121104), National Natural Science Foundation of China (No. U22B2059), the Anhui Natural Science Foundation (No. 2508085ZD006), National Natural Science Foundation of China (No.62502404), Hong Kong Research Grants Council (Research Impact Fund No.R1015-23, Collaborative Research Fund No.C1043-24GF, General Research Fund No. 11218325), Institute of Digital Medicine of City University of Hong Kong (No.9229503), Huawei (Huawei Innovation Research Program), Tencent (Tencent Rhino-Bird Focused Research Program, Tencent University Cooperation Project), Didi (CCF-Didi Gaia Scholars Research Fund), Kuaishou (CCF-Kuaishou Large Model Explorer Fund No. 2025008, Kuaishou University Cooperation Project), and Bytedance.

Limitations

Overall, our method is effective for improving long-horizon personalization memory by learning for more structured and consistent memory evolution. However, the second-stage optimization relies on an LLM-based scorer to provide guideline-aligned process rewards, which makes performance sensitive to scorer reliability. Moreover, our method requires careful tuning of the per-round token budget and the number of evolution rounds; when long histories are split into many rounds, small update errors can compound over time and lead to unintended forgetting or over-generalized memory entries. Finally, our current design treats memory evolution as a single-objective policy under a fixed guideline; extending it to explicitly balance multiple competing objectives (e.g., stability vs. plasticity, informativeness vs. brevity) remains non-trivial and may require additional control mechanisms.

Ethical considerations

Our method is a general memory-evolution framework intended to support personalized agents, and it primarily improves *how* existing memories are organized and optimized rather than expanding the scope of the LLM system's access. The primary ethical risk arises from misuse rather than from the method itself: if deployed without appropriate safeguards, persistent memory could be used to over-collect user information or to enable unwanted profiling. In responsible deployments, memory should follow data-minimization principles, avoid storing sensitive identifiers, and provide clear user controls for inspection, correction, and deletion; additionally, retention policies and access control should be enforced at the system level to ensure the system remains aligned with privacy expectations as application contexts evolve.

References

Joseph W Alba and Lynn Hasher. 1983. Is memory schematic? *Psychological Bulletin*, 93(2):203.

Nuo Chen, Hongguang Li, Jianhui Chang, Juhua Huang, Baoyuan Wang, and Jia Li. 2025a. Compress to impress: Unleashing the potential of compressive memory in real-world long-term conversations. In

Proceedings of the 31st International Conference on Computational Linguistics, pages 755–773.

Nuo Chen, Hongguang Li, Jianhui Chang, Juhua Huang, Baoyuan Wang, and Jia Li. 2025b. Compress to impress: Unleashing the potential of compressive memory in real-world long-term conversations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 755–773.

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

Yimin Deng, Yuqing Fu, Derong Xu, Yejing Wang, Wei Ni, Jingtong Gao, Xiaopeng Li, Chengxu Liu, Xiao Han, Guoshuai Zhao, and 1 others. 2026. Enhancing conversational agents via task-oriented adversarial memory adaptation. *arXiv preprint arXiv:2601.21797*.

Yiming Du, Hongru Wang, Zhengyi Zhao, Bin Liang, Baojun Wang, Wanjun Zhong, Zezhong Wang, and Kam-Fai Wong. 2024. Perltqa: A personal long-term memory dataset for memory classification, retrieval, and fusion in question answering. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 152–164.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, and 1 others. 2025. Lightmem: Lightweight and efficient memory-augmented generation. *arXiv preprint arXiv:2510.18866*.

Jingtong Gao, Bo Chen, Xiangyu Zhao, Weiwen Liu, Xiangyang Li, Yichao Wang, Wanyu Wang, Huifeng Guo, and Ruiming Tang. 2025. Llm4rerank: Llm-based auto-reranking framework for recommendations. In *Proceedings of the ACM on Web Conference 2025*, pages 228–239.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, and 28 others. 2025. Memory in the age of ai agents. *Preprint*, arXiv:2512.13564.

Pengyue Jia, Yiding Liu, Xiangyu Zhao, Xiaopeng Li, Changying Hao, Shuaiqiang Wang, and Dawei Yin. 2024. Mill: Mutual verification with large language models for zero-shot query expansion. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2498–2518.

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

Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh Rao Vijjini, Jiashu He, Hanchao Yu, and 1 others. 2025b. Personamem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory. *arXiv preprint arXiv:2512.06688*.

Seo Hyun Kim, Tzu-iunn Ong, Taeyoon Kwon, Namyoung Kim, Keummin Ka, SeongHyeon Bae, Yohan Jo, Seung-won Hwang, Dongha Lee, Jinyoung Yeo, and 1 others. 2024. Theanine: Revisiting memory management in long-term conversations with timeline-augmented response generation. *arXiv e-prints*, pages arXiv–2406.

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

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2024a. Hello again! llm-powered personalized agent for long-term dialogue. *arXiv preprint arXiv:2406.05925*.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, and 1 others. 2024b. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*.

Yuchen Li, Hengyi Cai, Rui Kong, Xinran Chen, Jiamin Chen, Jun Yang, Haojie Zhang, Jiayi Li, Jiayi Wu, Yiqun Chen, and 1 others. 2025a. Towards ai search paradigm. *arXiv preprint arXiv:2506.17188*.

Zhiyu Li, Shichao Song, Chenyang Xi, Hanyu Wang, Chen Tang, Simin Niu, Ding Chen, Jiawei Yang, Chunyu Li, Qingchen Yu, and 1 others. 2025b. Memos: A memory os for ai system. *arXiv preprint arXiv:2507.03724*.

Lei Liu, Xiaoyan Yang, Yue Shen, Binbin Hu, Zhiqiang Zhang, Jinjie Gu, and Guannan Zhang. 2023. Think-in-memory: Recalling and post-thinking enable llms with long-term memory. *arXiv preprint arXiv:2311.08719*.

Qidong Liu, Xiangyu Zhao, Yuhao Wang, Yejing Wang, Zijian Zhang, Yuqi Sun, Xiang Li, Maolin Wang, Pengyue Jia, Chong Chen, and 1 others. 2025. Large language model enhanced recommender systems: Methods, applications and trends. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6096–6106.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. 2025. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. *arXiv preprint arXiv:2508.09736*.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T Le, Samira Daruki, Xiangru Tang, and 1 others. 2025. Reasoningbank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2023. MemGPT: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. 2025. Secom: On memory construction and retrieval for personalized conversational agents. In *The Thirteenth International Conference on Learning Representations*.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 7957–7968.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. *Preprint*, arXiv:2409.05591.

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

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. 2024. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms. *arXiv preprint arXiv:2410.14052*.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

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

Juntao Tan, Liangwei Yang, Zuxin Liu, Zhiwei Liu, Rithesh RN, Tulika Manoj Awalgaonkar, Jianguo Zhang, Weiran Yao, Ming Zhu, Shirley Kokane, and 1 others. 2025a. Personabench: Evaluating ai models on understanding personal information through accessing (synthetic) private user data. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 878–893.

Zhen Tan, Jun Yan, I Hsu, Rujun Han, Zifeng Wang, Long T Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, and 1 others. 2025b. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. *arXiv preprint arXiv:2503.08026*.

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, and 1 others. 2025a. Agent kb: Leveraging cross-domain experience for agentic problem solving. *arXiv preprint arXiv:2507.06229*.

Xinyu Tang, Xiaolei Wang, Wayne Xin Zhao, Siyuan Lu, Yaliang Li, and Ji-Rong Wen. 2025b. Unleashing the potential of large language models as prompt optimizers: Analogical analysis with gradient-based model optimizers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25264–25272.

LangChain Team. 2023. Conversation summary memory.

Timothy J Teyler and Pascal DiScenna. 1986. The hippocampal memory indexing theory. *Behavioral neuroscience*, 100(2):147.

Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. 2025a. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*, page 130193.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788.

Yu Wang and Xi Chen. 2025. Mirix: Multi-agent memory system for llm-based agents. *arXiv preprint arXiv:2507.07957*.

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

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2025c. Agent workflow memory. In *Forty-second International Conference on Machine Learning*.

Qingsong Wen, Jing Liang, Carles Sierra, Rose Luckin, Richard Tong, Zitao Liu, Peng Cui, and Jiliang Tang. 2024. Ai for education (ai4edu): Advancing personalized education with llm and adaptive learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6743–6744, New York, NY, USA. Association for Computing Machinery.

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. 2025. From human memory to ai memory: A survey on memory mechanisms in the era of llms. *arXiv preprint arXiv:2504.15965*.

Derong Xu, Xinhang Li, Ziheng Zhang, Zhenxi Lin, Zhihong Zhu, Zhi Zheng, Xian Wu, Xiangyu Zhao, Tong Xu, and Enhong Chen. 2025a. Harnessing large language models for knowledge graph question answering via adaptive multi-aspect retrieval-augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25570–25578.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. 2026. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In *The Fourteenth International Conference on Learning Representations*.

Derong Xu, Ziheng Zhang, Zhenxi Lin, Xian Wu, Zhihong Zhu, Tong Xu, Xiangyu Zhao, Yefeng Zheng, and Enhong Chen. 2024. Multi-perspective improvement of knowledge graph completion with large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11956–11968, Torino, Italia. ELRA and ICCL.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.

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

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Kristian Kersting, Jeff Z Pan, Hinrich Schütze, and 1 others. 2025. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

Howard Yen, Tianyu Gao, and Danqi Chen. 2024. Long-context language modeling with parallel context encoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2588–2610.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and 1 others. 2025. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*.

Qianhao Yuan, Jie Lou, Zichao Li, Jiawei Chen, Yaojie Lu, Hongyu Lin, Le Sun, Debing Zhang, and Xianpei Han. 2025. Memsearcher: Training llms to reason, search and manage memory via end-to-end reinforcement learning. *arXiv preprint arXiv:2511.02805*.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. 2025. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639(8055):609–616.

Chao Zhang, Haoxin Zhang, Shiwei Wu, Di Wu, Tong Xu, Xiangyu Zhao, Yan Gao, Yao Hu, and Enhong Chen. 2025a. Notellm-2: Multimodal large representation models for recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 2815–2826.

Guibin Zhang, Muxin Fu, Kun Wang, Guancheng Wan, Miao Yu, and Shuicheng YAN. 2025b. G-memory: Tracing hierarchical memory for multi-agent systems. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Guibin Zhang, Muxin Fu, and Shuicheng Yan. 2025c. Memgen: Weaving generative latent memory for self-evolving agents. *arXiv preprint arXiv:2509.24704*.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita-Velez, Yue Liao, Hongru Wang, and 6 others. 2025d. The landscape of agentic reinforcement learning for llms: A survey. *Preprint*, arXiv:2509.02547.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, and 1 others. 2024a. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*.

Peiyan Zhang, Haibo Jin, Leyang Hu, Xinnuo Li, Liying Kang, Man Luo, Yangqiu Song, and Haohan Wang. 2024b. Revolve: Optimizing ai systems by tracking response evolution in textual optimization. *arXiv preprint arXiv:2412.03092*.

Shaokun Zhang, Jieyu Zhang, Jiale Liu, Linxin Song, Chi Wang, Ranjay Krishna, and Qingyun Wu. 2024c. Offline training of language model agents with functions as learnable weights. In *Forty-first International Conference on Machine Learning*.

Weizhi Zhang, Xinyang Zhang, Chenwei Zhang, Liangwei Yang, Jingbo Shang, Zhepei Wei, Henry Peng Zou, Zijie Huang, Zhengyang Wang, Yifan Gao, Xiaoman Pan, Lian Xiong, Jingguo Liu, Philip S. Yu, and Xian Li. 2025e. Personaagent: When large language model agents meet personalization at test time. In *First Workshop on Multi-Turn Interactions in Large Language Models*.

Yingyi Zhang, Pengyue Jia, Derong Xu, Yi Wen, Xianneng Li, Yichao Wang, Wenlin Zhang, Xiaopeng Li, Weinan Gan, Huifeng Guo, and 1 others. 2026a. Personalize before retrieve: Llm-based personalized query expansion for user-centric retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 16406–16414.

Yingyi Zhang, Junyi Li, Wenlin Zhang, Pengyue Jia, Xianneng Li, Yichao Wang, Derong Xu, Yi Wen, Huifeng Guo, Yong Liu, and Xiangyu Zhao. 2026b. Evoking user memory: Personalizing LLM via recollection-familiarity adaptive retrieval. In *The Fourteenth International Conference on Learning Representations*.

Yuxiang Zhang, Jiangming Shu, Ye Ma, Xueyuan Lin, Shangxi Wu, and Jitao Sang. 2025f. Memory as action: Autonomous context curation for long-horizon agentic tasks. *arXiv preprint arXiv:2510.12635*.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024d. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. 2025. Do llms recognize your preferences? evaluating personalized preference following in llms. *arXiv preprint arXiv:2502.09597*.

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.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. 2025. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *arXiv preprint arXiv:2506.15841*.

A GRPO for Memory Evolution

For completeness, we summarize the GRPO objective for the multi-turn memory evolution process. Following MemAgent (Yu et al., 2025), we optimize memory evolution over groups of trajectories. For a given input $(\mathcal{H}, x)$, the current policy π_ϕ generates a group of G trajectories $\{\tau_i\}_{i=1}^{G}$, with corresponding rewards $\{R_i\}_{i=1}^{G}$. In the multi-conversation setting, each trajectory τ_i is further decomposed into n_i conversations,

$$\tau_i = \{\tau_{i,1}, \tau_{i,2}, \ldots, \tau_{i,n_i}\},$$

where $\tau_{i,j}$ denotes the token sequence of the j-th conversation. GRPO normalizes rewards within each group and defines a group-relative advantage:

$$\widehat{A}_i = \frac{R_i - \mathrm{mean}(\{R_j\}_{j=1}^{G})}{\mathrm{std}(\{R_j\}_{j=1}^{G})}. \tag{7}$$

This advantage is then assigned to all token-level actions in τ_i, including both memory-update tokens and answer tokens across all conversations. Let $r_{i,j,t}(\phi)$ denote the importance-sampling ratio between the current policy and a frozen reference policy π_{ref} at token step t in the j-th conversation of trajectory τ_i. The multi-conversation GRPO objective is written as

$$J_{\text{GRPO}}(\phi) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{\sum_{j=1}^{n_i}|\tau_{i,j}|}\sum_{j=1}^{n_i}\sum_{t\in\tau_{i,j}} \min\Big(r_{i,j,t}(\phi)\widehat{A}_i, \mathrm{clip}\big(r_{i,j,t}(\phi), 1-\epsilon, 1+\epsilon\big)\widehat{A}_i\Big) - \beta\,\mathrm{KL}(\pi_\phi \,\|\, \pi_{\text{ref}})\Bigg]. \tag{8}$$

B Hyperparameter Settings

We summarize the hyperparameter settings for training and inference in Table 4.

C Datasets

C.1 PersonaMem Dataset

PersonaMem (Jiang et al., 2025a) is a large-scale benchmark for evaluating long-term personalization in conversational LLMs. It contains interaction histories for **20 simulated personas**, each designed with rich static attributes (e.g., demographics and occupation) and dynamic traits and preferences that evolve over time across **15 diverse**

Phase	Hyperparameters
Training (Stage 1)	Context = RAG Round size = 512 Optimization steps = $\{10, 30, 50, 70\}$ Temperature = 1.0 Top-p = 1.0 Max output tokens = 2048
Training (Stage 2)	Context = RAG Batch size = 4 Round size = 512 Learning rate = 1×10^{-6} Temperature = 1.0 Top-p = 1.0 Rollout batch size = 8 Rollout n = {2,4,8} Epochs = 5 Max output tokens = 2048
Inference	Context = Full history Round size = $\{1\text{K}, 2\text{K}, 4\text{K}, 8\text{K}, 16\text{K}\}$ Serving = vLLM Max output tokens = 2048 Temperature = 0.0

Table 4: Hyperparameter settings used for training and inference.

Statistic	PersonaMem		
Tokens per history	~32k	~128k	~1M
# QA pairs	589	2727	2674
# Sessions per history	10	20	60
Avg. # utterances	167.1	758.3	3607.9

Table 5: Statistics of the PersonaMem dataset at different context lengths. Token counts denote the approximate total context length per interaction history; utterance counts are averaged over histories.

real-world task domains such as food recommendation, travel planning, and therapy consultation. For every persona, multi-session conversations are constructed in which the user engages with a chatbot over **7 types of in-situ queries** that probe different personalization capabilities (e.g., recalling user facts, tracking preference evolution, and providing preference-aligned suggestions). Each session consists of 15-30 user–assistant turns, and histories are instantiated at three context scales by concatenating 10, 20, or 60 sessions, yielding approximate context lengths of 32k, 128k, and 1M tokens, respectively. At evaluation time, models must select appropriate responses to user queries conditioned on the interaction history, thereby testing their ability to evolve over dynamic user profiles. The main statistics of PersonaMem are summarized in Table 5.

Statistic	Value
Explicit queries	1,000
Implicit queries	1,000
Maximum inserted conversations	24
Maximum inserted turns	326
Avg. turns / conversation	13.58
Total tokens	108,102
Avg. tokens / conversation	4,504.25

Table 6: Dataset statistics for PrefEval multiple-choice classification.

User	Queries	Corpus	Conv.	AI	E-com.
1	48	110	84	23	3
2	43	90	78	8	4
3	42	64	51	12	1
4	46	85	71	14	0
5	44	84	59	21	4
6	40	94	79	14	1
Sum	263	527	422	92	13

Table 7: Statistics of the PersonaBench subset across six users. *Corpus* is the sum of *Conv.*, *AI*, and *E-com.*.

C.2 PrefEval Dataset

PrefEval is a long-context, multi-session benchmark for evaluating whether LLMs can infer, retrieve, and act on user preferences in realistic conversational settings, with an emphasis on four aspects: preference inference, long-context retrieval, preference following, and personalization proactiveness. The dataset comprises 1,000 unique preference–query pairs, and spans 20 everyday topics grouped into seven domains: *Entertainment* (Shows, Music & Books, Sports, Games), *Travel* (Activities, Restaurant, Hotel, Transport), *Lifestyle* (Dietary, Beauty, Fitness, Health), *Shopping* (Home, Fashion, Motors, Technology), *Education* (Resources, Learn Styles), *Professional Ownership*, and *Professional Work Style*. PrefEval supports two evaluation formats: a free-form generation setting and a 4-way multiple-choice classification setting in which exactly one option is consistent with the stated preference. To stress long-range personalization, the benchmark inserts unrelated multi-session dialogue turns between the preference revelation and the final query. In our experiments, we use 1,000 explicit and 1,000 implicit instances under the multiple-choice classification setting, and insert 50 intervening turns as distractor context; summary statistics of our subset are reported in Table 6.

C.3 PersonaBench Dataset

PersonaBench (Tan et al., 2025a) is a benchmark designed to evaluate personalized retrieval and question answering grounded in user-specific context. For each user, it provides a heterogeneous personal corpus comprising (i) conversations with friends (*Conv.*), (ii) dialogues with AI assistants (*AI*), and (iii) e-commerce purchase histories (*E-com.*). The evaluation queries are typically short and underspecified, requiring models to resolve implicit intent by grounding responses in evidence distributed across the user's historical interactions and behaviors. This setting tests a model's ability to align with diverse, user-dependent semantics under realistic contextual ambiguity. Table 7 summarizes the per-user query counts and corpus statistics for the six-user subset used in our experiments.

D Comparison in Different Categories

Comparison in Different Categories of PersonaMem. Table 8 reports category-wise results on PersonaMem under 32K and 128K interaction histories. Across both scales, **MemCoE** achieves the best overall performance (57.06 at 32K; 47.24 at 128K), and the gains are concentrated on memory-dependent personalization abilities. On 32K histories, **MemCoE** leads in Recall facts (59.50), Prefs evolve (68.64), Update reasons (81.25), Aligned recs (62.22), and New Scenarios (56.52), which jointly drives a clear margin over the strongest baselines (e.g., 57.06 vs. 53.58 for MemAgent). When scaling to 128K, *Long Context* degrades sharply overall (25.05), while memory-based methods remain substantially stronger; within them, **MemCoE** stays best-performing and ranks first on Latest prefs (54.44), Prefs evolve (66.47), Aligned recs (51.61), and New Scenarios (38.35). In contrast, Suggest ideas is not a strength for **MemCoE** (8.86/11.68), where methods that do not emphasize memory evolution (e.g., Mem0 or Long Context) are higher, indicating that our improvements primarily come from better tracking and applying evolving persona preferences rather than open-ended QA. Overall, the category-wise gains suggest that explicitly *structuring* memory operations and then *selecting* what to keep is most effective for preference-heavy queries, and the advantage becomes more pronounced as the interaction history grows longer.

Comparison in Different Categories of PrefEval. Table 9 breaks down PrefEval performance by domain under **Explicit** and **Implicit** preference settings. Under Explicit Memory, **MemCoE** attains the best overall accuracy (81.30) and shows consistently strong gains on preference-heavy domains, ranking first on Travel (82.24), Lifestyle (84.83), Shop (82.11), and Education (85.88), while remaining competitive on **Entertain** (76.92) and Professional (83.33). A notable exception is Pet, where **MemCoE** (67.44) trails MemAgent (74.42), suggesting that not all topics benefit equally from the same memory update behavior. Under Implicit Memory, the task becomes more challenging for all methods, yet **MemCoE** again leads overall (69.90) and improves most clearly on domains that require inferring latent preferences from context, including Lifestyle (73.93), Shop (72.63), Education (70.59), and Professional (63.89). Compared with memory-bank baselines (Mem0/A-Mem/LightMem), the advantage of **MemCoE** is broad across domains in both settings, indicating that it better resists long-range distractors and preserves preference-relevant signals. Overall, these domain-wise results reflect PrefEval's construction: the inserted unrelated turns make long-range preference retrieval and faithful preference following the main bottlenecks, and **MemCoE** improves most on domains where precise preference identification and consistent application are essential.

Comparison in Different Categories of PersonaBench. Table 10 reports category-wise macro F1 on PersonaBench under increasing noise in the memory bank, where queries are short and underspecified, and evidence is distributed across heterogeneous user corpora (Table 7). Without noise, **MemCoE** achieves the best overall score (32.27) and is particularly strong on preference- and interaction-driven categories, leading on Pref. (Hard) (37.02) and Social (38.95), while remaining competitive on Pref. (Easy) (34.61); in contrast, Basic Info favors direct long-context inclusion (Long Context: 29.23 vs. **MemCoE**: 26.74), suggesting that simple factual lookup is less dependent on selective memory evolution. As noise increases, all methods degrade, but **MemCoE** remains the best overall method at every noise level (29.89 at 0.3; 25.99 at 0.5; 25.09 at 0.7), indicating stronger robustness to irrelevant or misleading memory entries. The category breakdown further shows that **MemCoE** maintains clear advantages on Pref. (Easy) under heavier noise (36.91 at 0.7), while the performance gap on Pref. (Hard) and Social narrows against strong baselines (e.g., A-Mem), reflecting that fine-grained prefer-

Method	Recall facts	Suggest ideas	Latest prefs	Prefs evolve	Update reasons	Aligned recs	New Scenarios	Overall
▼ *32K memory corpus data*								
Long Context	28.93	11.39	-	44.07	61.25	35.56	15.22	34.36
RAG	42.98	15.19	-	59.32	80.00	51.11	36.96	48.67
Mem0	47.93	**19.41**	-	46.61	79.58	57.04	42.75	48.53
A-Mem	47.11	10.13	-	61.86	80.00	44.44	30.43	48.26
LightMem	52.07	10.13	-	65.25	77.50	51.11	32.61	50.72
MemAgent	54.55	8.86	-	65.25	76.25	57.78	54.35	53.58
Mem-α	57.02	7.59	-	64.41	72.50	60.00	54.35	53.37
MemCoE (Ours)	**59.50**	8.86	-	**68.64**	**81.25**	**62.22**	**56.52**	**57.06**
▼ *128K memory corpus data*								
Long Context	18.12	23.36	20.62	38.62	39.77	21.70	16.99	25.05
RAG	**54.37**	**19.67**	36.45	52.69	58.71	41.94	29.61	38.90
Mem0	56.25	20.49	37.77	55.09	57.20	41.64	29.13	39.67
A-Mem	36.88	16.19	38.73	59.88	62.50	35.19	28.16	38.22
LightMem	40.00	17.42	42.33	61.08	**64.02**	35.48	25.73	39.93
MemAgent	50.62	10.86	49.40	61.68	59.47	47.80	35.44	43.59
Mem-α	48.75	10.45	50.12	59.28	54.55	47.21	36.89	42.86
MemCoE (Ours)	52.50	11.68	**54.44**	**66.47**	**64.02**	**51.61**	**38.35**	**47.24**

Table 8: Category-wise accuracy (%) on PersonaMem under 32K and 128K interaction histories. "–" indicates the category is not available in the dataset.

Method	Travel	Entertain	Lifestyle	Shop	Education	Professional	Pet	Overall
▼ *Explicit Memory*								
Long Context	31.78	30.77	32.70	30.00	32.94	27.78	39.53	31.70
RAG	45.33	52.04	47.87	45.79	42.35	58.33	48.84	47.80
Mem0	58.41	61.99	57.35	53.68	54.12	69.44	46.51	57.60
A-Mem	62.15	69.23	60.19	61.05	54.12	66.67	55.81	62.30
LightMem	65.42	68.33	65.40	62.11	56.47	66.67	53.49	64.20
MemAgent	71.03	**77.83**	70.62	71.05	62.35	**83.33**	**74.42**	72.30
Mem-α	78.50	73.76	67.30	70.00	65.88	77.78	67.44	71.90
MemCoE (Ours)	**82.24**	76.92	**84.83**	**82.11**	**85.88**	**83.33**	67.44	**81.30**
▼ *Implicit Memory*								
Long Context	30.84	27.60	34.12	29.47	34.12	25.00	34.88	30.80
RAG	26.17	41.18	33.65	29.47	30.59	13.89	44.19	32.40
Mem0	43.93	51.13	48.34	42.11	44.71	30.56	60.47	46.40
A-Mem	51.40	55.20	55.45	47.37	54.12	50.00	58.14	52.80
LightMem	50.47	60.63	58.77	51.58	47.06	44.44	65.12	54.80
MemAgent	59.35	66.52	65.88	63.68	56.47	52.78	**81.40**	63.60
Mem-α	61.68	66.52	62.09	61.05	60.00	52.78	67.44	62.50
MemCoE (Ours)	**64.02**	**69.23**	**73.93**	**72.63**	**70.59**	**63.89**	74.42	**69.90**

Table 9: Domain-wise accuracy (%) on PrefEval multiple-choice classification under Explicit vs. Implicit preference.

ence grounding and social inference are the most sensitive to noisy evidence. Overall, the results highlight that **MemCoE** is most beneficial when personalization requires resolving implicit intent over long, heterogeneous histories, and it degrades more gracefully as the memory bank becomes noisier.

E Additional Experiments

E.1 Comparison with Post-Training Baselines

Table 11 compares **MemCoE** with standard post-training baselines trained on the same 300 PersonaMem training data, where the baselines do not perform memory evolution and instead optimize question answering directly. Moving from Frozen to SFT and then to PPO/GRPO yields steady overall improvements (34.36→46.83→49.49/50.31), indicating that post-training helps, but the gains are uneven across personalization skills. In contrast, **MemCoE** achieves the best overall score (57.06) and leads on most categories, including Recall facts (59.50), Prefs evolve (68.64), Aligned recs (62.22), and New Scenarios (56.52). This pattern is consistent with our design: by explicitly internalizing memory extraction, update, and forgetting into a

Method	Basic Info	Pref. (Easy)	Pref. (Hard)	Social	Overall	Basic Info	Pref. (Easy)	Pref. (Hard)	Social	Overall
	▼ *Without Noise Memory*					▼ *With 0.3 Noise Memory*				
Long Context	**29.23**	34.41	24.51	29.36	29.00	21.15	22.24	18.78	13.55	19.10
RAG	24.32	34.06	32.77	33.68	29.09	26.14	39.29	28.63	26.53	28.16
Mem0	19.23	19.09	18.80	12.58	17.60	17.41	29.08	21.87	18.39	19.75
A-Mem	25.69	34.04	34.46	34.90	30.32	27.00	**39.65**	26.94	**27.61**	28.56
LightMem	18.92	20.41	21.40	16.96	19.08	21.25	22.60	18.52	11.82	18.74
MemAgent	14.32	31.09	26.58	21.48	20.05	16.30	32.36	18.76	19.77	19.36
Mem-α	14.91	25.92	30.03	19.55	19.92	11.87	27.71	26.00	15.52	17.02
MemCoE	26.74	**34.61**	**37.02**	**38.95**	**32.27**	**29.81**	35.12	**29.92**	27.48	**29.89**
	▼ *With 0.5 Noise Memory*					▼ *With 0.7 Noise Memory*				
Long Context	18.65	16.52	17.90	16.70	17.83	11.25	22.26	18.62	7.75	13.00
RAG	22.45	27.49	22.58	27.95	24.31	18.38	31.83	25.85	26.05	23.00
Mem0	18.33	24.31	23.79	15.04	19.22	15.07	21.38	21.58	18.76	17.80
A-Mem	22.28	23.64	**28.12**	**29.73**	25.19	18.42	29.96	**28.11**	31.42	24.45
LightMem	22.85	20.47	17.09	14.59	19.65	20.30	19.80	17.35	12.00	17.80
MemAgent	13.40	16.73	21.13	19.27	16.51	13.76	26.45	23.01	18.44	17.92
Mem-α	11.20	17.31	22.36	22.28	16.43	12.42	22.66	18.55	16.42	15.59
MemCoE	**23.82**	**33.01**	23.19	29.21	**25.99**	**20.62**	**36.91**	27.09	**27.00**	**25.09**

Table 10: Category-wise macro F1 (%) on PersonaBench for the six-user subset under different noise rates injected into the memory bank.

Method	Recall facts	Suggest ideas	Latest prefs	Prefs evolve	Update reasons	Aligned recs	New Scenarios	Overall
Frozen	28.93	11.39	-	44.07	61.25	35.56	15.22	34.36
SFT	42.15	**15.19**	-	55.93	**81.25**	48.89	28.26	46.83
PPO	51.24	**15.19**	-	60.17	75.00	44.44	36.96	49.49
GRPO	52.07	16.46	-	63.56	71.25	46.67	36.96	50.31
MemCoE	**59.50**	8.86	-	**68.64**	**81.25**	**62.22**	**56.52**	**57.06**

Table 11: Comparison with different post-training methods on PersonaMem 32K memory corpus data.

dedicated memory-evolution mechanism, **MemCoE** improves long-horizon preference tracking and generalization beyond what QA-only post-training captures.

E.2 Evaluation of Preference Retention

Figure 7 directly tests whether our method can preserve useful user preference information inside the memory bank throughout multi-round memory evolution. We use PrefEval (Explicit) because the preference is inserted at the beginning (round 0), which makes retention measurable and avoids ambiguity about when the preference should appear. After inserting the preference, we run multi-round evolution with a fixed 4K-token context per round, and ask `Gemini-2.5-Pro` to judge whether the memory bank still contains the inserted preference. At round 0, both methods have 100% retention by construction, after which their retention curves diverge rapidly as rounds increase: MemAgent exhibits a steep and nearly monotonic decay, dropping to roughly 51% retention by round 10, whereas our method degrades much more slowly and remains around 74% at round 10. This yields a substantially smaller absolute decrease in retention for our method (about 26%) compared to MemAgent (about 49%), and the growing shaded gap indicates that the advantage accumulates over time rather than being a one-off effect. Qualitatively, this behavior aligns with our design goal. Specifically, by introducing an induced memory-update guideline to regulate what to keep, refine, or delete, our evolution process is less prone to overwriting or dilution of the initially injected preference under long interaction histories. In contrast, MemAgent appears more vulnerable to preference drift and forgetting as the number of rounds increases, since later interactions can introduce competing signals and noisy content that interfere with the original preference.

E.3 Error Analysis

Figure 8 breaks down errors according to whether they originate from the memory evolution stage

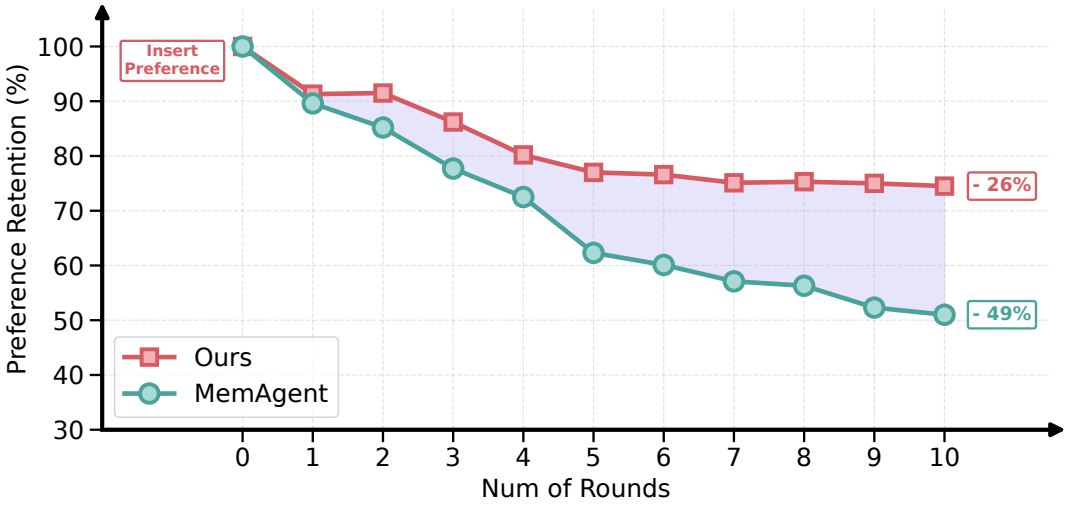


Figure 7: Preference retention during multi-round memory evolution on PrefEval (Explicit). We insert a user preference at round 0 and then run memory evolution for subsequent rounds, where each round uses a 4K-token dialogue context. A strong judge model (`Gemini-2.5-Pro`) verifies whether the preference remains in the memory bank after each round.

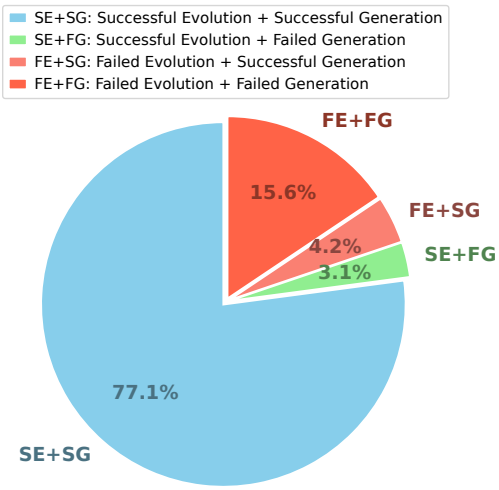


Figure 8: Error analysis decomposing failures into the memory evolution stage and the response generation stage. We use the same setup as Figure 7, where successful evolution means the memory bank captures the user preference, and successful generation means the final answer is correct given the evolved memory.

or the response generation stage under the same preference-retention setup as Figure 7. The dominant category is SE+SG (Successful Evolution + Successful Generation) at 77.1%, indicating that in most cases the system both captures the user preference in the memory bank and leverages it to produce a correct response. The remaining 22.9% errors are split across three failure modes: SE+FG accounts for 3.1%, showing that even when the preference is correctly stored, generation can still fail to use it; FE+SG accounts for 4.2%, suggesting that correct answers can occasionally be produced despite imperfect preference capture (e.g., the model may rely on residual context cues rather than the memory bank); and FE+FG accounts for 15.6%, which is the largest failure category and highlights that missed or incorrect preference capture during memory evolution often cascades into downstream response failures. Overall, this decomposition suggests that improving the reliability of the evolve stage, i.e., making it more consistent in capturing and preserving preferences, should yield the largest payoff. The comparatively small SE+FG slice indicates that, although present, generation errors are not the primary bottleneck in this setting.

E.4 Effect of Training Steps on RL-Based Baselines

To verify that the performance gap between MemCoE and RL-based baselines is not attributable to insufficient baseline training, we conduct a training-step study under an identical data budget and exper-

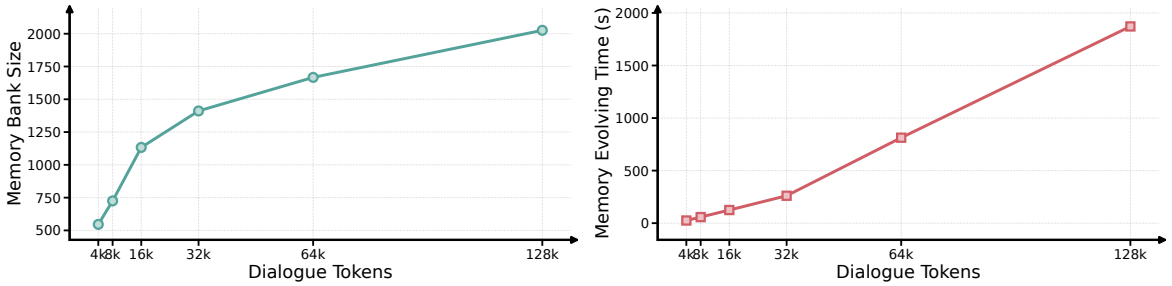
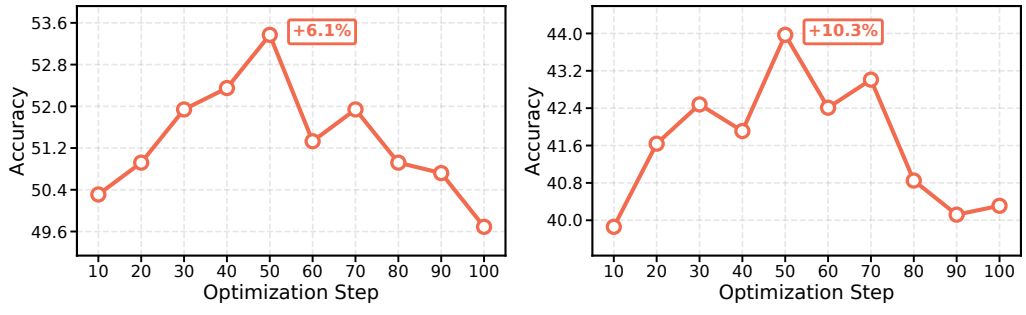
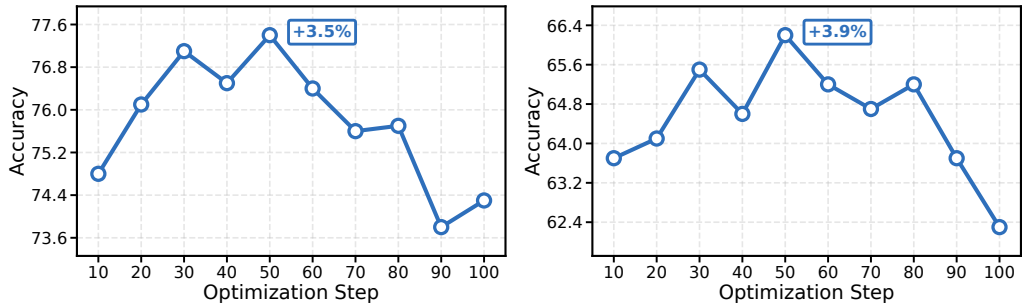


Figure 9: Scaling analysis on PersonaMem (128K). We increase the total dialogue tokens (each evolution round processes 4K tokens) and report the resulting memory bank size (left) and memory evolving time (right).



(a) PersonaMem (32K)

(b) PersonaMem (128K)



(c) PrefEval (Explicit)

(d) PrefEval (Implicit)

Figure 10: Hyperparameter analysis of optimization steps. The relative gains ($+\Delta\%$) are computed w.r.t. the performance at step 10.

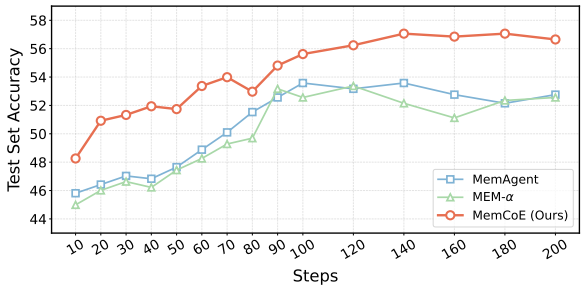


Figure 11: Test set Accuracy of RL-based baselines across RL training steps.

imental setting: 300 sampled PersonaMem training examples, batch size 4, and evaluation on PersonaMem-32K, varying only the number of RL update steps up to 200. As shown in Figure 11, both MemAgent and MEM-α reach a clear performance plateau around steps 100–140, with MemAgent peaking at 53.58 (steps 100/140) and MEM-α peaking at 53.37 (step 120), after which neither baseline exhibits consistent improvement—confirming that both models have converged within the given budget. Despite this, MemCoE continues to outperform the best checkpoints of both baselines by a substantial margin throughout training, demonstrating that the observed accuracy gap stems primarily from MemCoE's architectural design rather than any insufficiency in baseline training.

E.5 Scaling Analysis

Figure 9 analyzes how our memory system scales with longer dialogue context on PersonaMem

(128K) using 4K tokens per evolution round. As the dialogue tokens grow from 4K to 128K, the memory bank size increases from roughly 500 to around 2,000, and the curve is sublinear: it grows faster in the short-context regime and then gradually flattens as the history lengthens, which is consistent with consolidating stable information while removing redundant or outdated content to **keep memory overhead under control**. Meanwhile, the memory evolving time increases smoothly with dialogue length and follows an **approximately linear trend**, remaining well-behaved across the entire range; this indicates that the computational cost scales predictably with the amount of dialogue processed per round, with the slowly expanding memory bank introducing only a mild additional overhead in longer contexts.

E.6 Optimization Step Analysis

Figure 10 studies the number of optimization steps used in **MGI** for Memory Guideline Induction. Across all four settings, performance improves from the initial configuration and reaches a clear peak at an intermediate step budget, with the best results achieving relative gains of +6.1% (PersonaMem 32K), +10.3% (PersonaMem 128K), +3.5% (PrefEval Explicit), and +3.9% (PrefEval Implicit) compared to the 10-step setting. When the step count is too small, the guideline updates are likely under-developed: the aggregated textual gradients have limited opportunity to accumulate recurring error patterns across batches, so the induced guideline remains close to the initial prompt and cannot consistently regulate memory operations. Conversely, when the step count becomes large, the curves show a downward trend after the peak, suggesting diminishing returns and instability: repeated natural-language edits can over-specialize the guideline to feedback from later batches, or amplify small contradictions across textual gradients, which in turn weakens its ability to **generalize across histories and query types**. Overall, the results indicate that **MGI** benefits from enough iterations to consolidate batch-level critiques into a robust global policy, but requires a moderate step budget to avoid drifting away from broadly useful memory-update principles.

E.7 Comparison with TextGrad

To further validate MGI, we compare against TextGrad (Yuksekgonul et al., 2025), a strong general-purpose prompt optimizer, under the PersonaMem (32K) setting. Results are shown in Table 12: TextGrad improves over the manual prompt but still lags behind MGI by a substantial margin. This gap highlights that general prompt optimization is insufficient for our memory-evolution setting, where updates must be grounded in long-horizon trajectories. The results support that MGI's trajectory-grounded contrastive signal and batch aggregation are critical for inducing a high-quality memory guideline.

Table 12: Comparison with prompt optimization methods on PersonaMem (32K). * indicates statistically significant improvement over the second-best baseline (two-sided t-test, $p < 0.05$).

Method	PersonaMem (32K)
Manual Prompt	48.25 $\pm$ 0.68
TextGrad	49.83 $\pm$ 0.83
MemCoE (w/ only MGI)	**53.28** $\pm$ 0.76*

F Case Study

Figures 12–15 show four representative cases from PersonaMem and PersonaBench. In the two PersonaMem multiple-choice examples (Figures 12 and 13), **MemCoE** selects the ground-truth option (**(d)** and **(a)**), whereas all baselines choose different options, indicating that they fail to preserve or exploit the preference-relevant evidence needed for preference-aligned recommendations. In the two PersonaBench factual QA examples (Figures 14 and 15), **MemCoE** correctly outputs the user's age (**39**) and work location (**University**), while baselines frequently return *Unknown/Not specified* or claim missing information, consistent with information being unavailable at answer time due to memory evolution and/or retrieval failures.

G Prompts

G.1 Meta Prompt for Guideline Optimization

We implement a three-stage meta-prompt pipeline to optimize the guideline prompt used for memory evolution. As shown in Figure 16, `TEMPLATE_LOSS` performs a contrastive diagnosis by comparing a `correct_sample` and a `wrong_sample`, identifying why the correct update succeeds, why the incorrect update fails, and what systematic issues exist in `template_evolve`. Based on multiple such analyses, Figure 17 uses `TEMPLATE_AGGR` to aggregate diverse feedback into a single coherent sum-

CASE STUDY (PersonaMem-1): Preference-Aligned Recommendation (MCQ)

User question: I'm planning a weekend getaway and want to try something creatively fulfilling. What would you suggest?

Question type: *provide_preference_aligned_recommendations*

Topic: *musicRecommendation*

Options:
(a) How about spending the weekend learning traditional island cooking techniques or delving into the art of Polynesian tattoo design? Imagine mastering the skills of preparing intricate island feasts or crafting culturally rich tattoos on a canvas. This experience would not only honor your heritage but also keep those traditions alive through your newfound expertise. It's a fantastic way to celebrate your roots and create something truly meaningful!
(b) Why not explore the vibrant world of painting by setting up an easel in an art studio or your backyard? Picture yourself dipping brushes into vivid colors, creating masterpieces on canvas inspired by your surroundings. This artistic journey allows you to express emotions through visual art and discover new techniques that can be both relaxing and rewarding, offering a delightful escape from the ordinary.
(c) Consider indulging in the art of storytelling by crafting a compelling narrative or writing poetry in a cozy nook. Envision weaving intricate tales or rhythmic verses that captivate the mind and soul, drawing inspiration from your own experiences or the world around you. This literary venture not only sharpens your writing skills but also provides a channel for introspection and endless creativity.
(d) How about diving into a soundscape adventure by capturing the symphony of nature in an enchanting forest or by a tranquil lake? Imagine blending the serene rustling of leaves, the melodic rush of water streams, and the rhythmic patter of rain to craft a unique auditory tapestry. This will not only ignite your creativity but also allow you to relive those serene landscapes through your sound engineering skills. It's a perfect way to reconnect with nature's music and create something truly spectacular!

Correct answer: (d)

Predictions: Ours: (d); MemAgent: (a); Mem-α: (a); A-Mem: (a); LightMem: (a); Mem0: (c).

Figure 12: PersonaMem case study (MCQ). **Our method** matches the ground-truth option **(d)**, while all baselines select different options.

CASE STUDY (PersonaMem-2): Suggesting New Ideas for Music Expression (MCQ)

User question: How can I find a more fulfilling way to express my love for music?

Question type: *suggest_new_ideas*

Topic: *musicRecommendation*

Options:
(a) You might consider exploring different avenues like writing about your musical journey or experimenting with performing live in settings that inspire you. Also, giving yourself the freedom to simply enjoy music without external pressures could rekindle your passion.
(b) Exploring sound engineering might offer a fulfilling way to express your love for music. Like one user who was inspired by a chance meeting with an audio engineer at a festival, you could dive into the world of creating digital music remixes. By blending different influences and styles, you may find inspiration in exploring the nuances of sound capturing and creating unique auditory landscapes that surpass traditional music boundaries.
(c) Collaborating with others who share your musical interests can also be a rewarding path. A user found fulfillment through working with musicians from diverse backgrounds, mixing traditional and electronic elements to expand their creative horizons. Such group efforts can enhance not only your musical explorations but also build lasting relationships, as you contribute unique perspectives to a collective musical vision.
(d) Consider getting involved in music criticism by writing album reviews. As highlighted by a diligent user after attending a workshop, reviews can be instrumental in shaping listener perspectives and enhancing understanding. By delving deeper into the contexts and intentions behind albums, you can enrich your musical experience and articulate your insights, potentially helping others to connect more deeply with the music.

Correct answer: (a)

Predictions: Ours: (a); MemAgent: (c); Mem-α: (c); A-Mem: (b); LightMem: (d); Mem0: (c).

Figure 13: PersonaMem case study (MCQ). **Our method** selects the correct option **(a)**, whereas each baseline chooses a different option.

CASE STUDY (PersonaBench-1): Basic User Fact (Open-form QA)

User question: At what age am I right now?
Question type: *Basic information*

Correct answer: 39
Predictions:
Ours: 39;
MemAgent: Not specified;
Mem-α: Not specified;
A-Mem: Unknown;
LightMem: Information not provided;
Mem0: Not specified.

Figure 14: PersonaBench case study (open-form QA). **Our method** outputs the correct age (**39**); baselines answer with missing/unknown information.

CASE STUDY (PersonaBench-2): Work Location (Open-form QA)

User question: What is the address of my work location?
Question type: *Basic information*

Correct answer: University
Predictions:
Ours: University;
MemAgent: Unknown;
Mem-α: Not specified;
A-Mem: Not specified;
LightMem: The provided information does not include the address of your work location.;
Mem0: There is no relevant information provided.

Figure 15: PersonaBench case study (open-form QA). **Our method** recovers the correct work location (**University**), while baselines report missing or unknown information.

mary, resolving inconsistencies and retaining the most consistent actionable points. Figure 18 uses `TEMPLATE_OPTIM` to revise `template_evolve` by following the aggregated feedback while preserving the required placeholders, producing an instruction prompt for guideline optimization.

G.2 Prompt for Memory Evolution and Final Answer Generation

To enable long-horizon personalization, we first prompt the model to evolve a structured user memory profile from newly observed dialogue chunks under evidence-bounded extraction and conflict-aware consolidation (Fig. 19). Notably, this prompt is progressively optimized rather than manually fixed: starting from a generic read & update instruction, it gradually evolves into a more constrained memory policy that emphasizes recency and usefulness, and finally enforces evidence-bounded updates, explicit conflict resolution, and selective exclusion of privacy-sensitive or unsupported content. This evolution is motivated by two recurring failure modes observed during optimization: unresolved conflicts can lead to inconsistent memory, while over-collection of one-off details can make the profile noisy and less useful for personalization.

For fair comparison, we then adopt shared final-answer prompts across all compared methods. As shown in Fig. 20, for multiple-choice benchmarks (PersonaMem, and PrefEval), the template instructs the model to select the most appropriate option based on user preferences in memory and to output only the option letter, enforcing a strict and comparable output format. For PersonaBench, which requires open-form answers, we use a separate template (Fig. 21) that constrains the output to only the name(s) of the relevant entity/entities and explicitly avoids any additional explanation, thereby stan-

TEMPLATE_LOSS: Contrastive Feedback for Memory Update

TEMPLATE_LOSS = """
Below are two examples of memory updates: one labeled as `correct_case` and the other as `wrong_case`.
These examples illustrate the process of updating memory blocks based on a user's question using `template_evolve` by querying a Large Language Model.

Your task is to apply **contrastive learning principles** to thoroughly compare the correct and incorrect samples.
Analyze the reasons why the **correct sample is successful** and identify the factors contributing to the **failure of the wrong sample**.
Critically evaluate the issues present in `template_evolve`, and propose how it can better capture **user preferences**.
Provide **actionable suggestions and feedback** for optimizing the template.

<correct_sample>
{correct_sample}
</correct_sample>

<wrong_sample>
{wrong_sample}
</wrong_sample>

 # Definitions:
- `user_question_or_message`: Represents the question or message provided by the user.
- `correct_answer`: Denotes the correct answer to the given question.
- `model_response`: Indicates the model's response, derived by combining the `user_question_or_message` with the `memory_bank`.
- `memory_bank`: Refers to a storage area that is updated during the memory refinement process.

 # The template requiring further refinement and optimization is as follows:

<template_evolve>
{template_evolve}
</template_evolve>

 # Please provide suggestions and feedback on how this template can be improved and optimized.
Do not directly update the template content; focus solely on providing recommendations.
"""

Figure 16: Meta prompt for generating contrastive feedback by comparing a correct and an incorrect memory-update case, and for diagnosing weaknesses in `template_evolve` (TEMPLATE_LOSS).

TEMPLATE_AGGR: Feedback Aggregation into a Coherent Summary

TEMPLATE_AGGR = """ **You are provided with multiple feedback responses from different analyses.** Your task is to **synthesize these into a single, coherent feedback summary.** Ensure that the final feedback:
1. **Captures the key insights** from each response.
2. **Resolves any conflicting points** by identifying the most consistent and relevant information.
3. Provides a **clear and concise summary** that reflects the overall consensus of the feedback.

Feedback Responses:
{feedback_responses}

Final Feedback Summary:
"""

Figure 17: Meta prompt for synthesizing multiple analysis outputs into a single consolidated feedback summary (TEMPLATE_AGGR).

dardizing response granularity and ensuring fair evaluation under identical prompting conditions.

TEMPLATE_OPTIM: Template Refinement Under Placeholder Constraints

TEMPLATE_OPTIM = """
Please refine and optimize the following instruction template `template_evolve` based on the provided feedback.

 # CRITICAL RULES:
 # 1. You MUST preserve the following placeholder tokens in the template:
For `template_evolve_new`: {memory}, {chunk}
These placeholders are **REQUIRED** and must appear exactly as shown (with curly braces) in your output.
If any of these placeholders are missing, the template will fail to work.
 # 2. Do NOT add any new placeholders. Only use the placeholders listed above.
Output the optimized prompt text directly without introducing any additional placeholder tokens.

 # Below is the template that needs to be revised and improved:

<template_evolve>
{template_evolve}
</template_evolve>

 # Please comprehensively adhere to the feedback provided below to update the template:

<feedback>
{feedback}
</feedback>

 # Place the optimized version of the template within the following tags:

<template_evolve_new>
</template_evolve_new>
"""

Figure 18: Meta prompt for updating `template_evolve` using aggregated feedback while strictly preserving required placeholders and forbidding new ones (TEMPLATE_OPTIM).

TEMPLATE_EVOLVE_STEP50: Evidence-Bound Memory Update from a New Chunk

TEMPLATE_EVOLVE_STEP50 = """

You are given a question with options, some new memory, and previous user memory. Read the new section and update the user memory by prioritizing **recent and relevant information** that aligns with the user's **preferences and experiences**.

<memory> {memory} **</memory>**

<section> {chunk} **</section>**

Update rules:
- **Evidence-bound:** Extract candidate memory items from chunk. Every stored item must be **directly supported** by chunk.
- **Relevance & stability:** Store **long-term preferences**, **stable facts**, **recurring habits**, **long-term goals**, and **interaction preferences**. Do **not** store one-off details (e.g., transient locations, momentary moods) unless explicitly stated as long-term.
- **Conflict handling:** If new info contradicts existing memory, prefer the **most recent supported** info as active, and mark the older one as **deprecated** with a short reason.
- **Privacy:** Do **NOT** store highly sensitive or uniquely identifying data (exact address, account credentials, financial/medical specifics, etc.).
- **No domain bias:** Do not assume any specific hobbies or interests unless stated in chunk.
- If chunk contains explicit user corrections/ratings about previous outputs, store them under **"interaction_feedback"**. Otherwise, do not create feedback entries.
- Merge into a **structured memory profile**. Keep it concise, non-redundant, and internally consistent.

"""

Figure 19: Meta prompt for updating the user memory profile from a newly observed dialogue chunk with evidence-bounded extraction and conflict-aware consolidation (TEMPLATE_EVOLVE_STEP50).

TEMPLATE_FINAL (MCQ): PersonaMem / PrefEval

TEMPLATE_FINAL = """ **You are presented with a question and its corresponding options.** Find the most appropriate option to the question based on **user preference in memory** and give your final answer (a), (b), (c), or (d). Put the answer in **\boxed{}**.

<question> {question} **</question>**

<options> {options} **</options>**

<memory> {memory} **</memory>**

Your answer:
"""

Figure 20: Shared prompt for final answer generation on multiple-choice benchmarks (PersonaMem, and PrefEval).

TEMPLATE_FINAL_OPEN (Open QA): PersonaBench

TEMPLATE_FINAL_OPEN = """ **You are provided with the following relevant information about a user:**

<memory> {memory} **</memory>**

Answer the question below as **directly and concisely as possible**, using only the **name(s) of the relevant entity or entities**. **Avoid adding any extra words or explanations.**

<question> {question} **</question>**

Your answer:
"""

Figure 21: Shared prompt for final answer generation on PersonaBench, constraining outputs to only the relevant entity name(s).