Controllable Memory Usage: Balancing Anchoring and Innovation in Long-Term Human-Agent Interaction

Anonymous ACL submission

Abstract

As LLM-based agents are increasingly used in long-term interactions, cumulative memory is critical for enabling personalization and maintaining stylistic consistency. However, most existing systems adopt an "all-or-nothing" approach to memory usage: incorporating all relevant past information can lead to *Memory Anchoring*, where the agent is trapped by past interactions, while excluding memory entirely results in under-utilization and the loss of important interaction history. We show that an agent's reliance on memory can be modeled as an explicit and user-controllable dimension. We first introduce a behavioral metric of memory dependence to quantify the influence of past interactions on current outputs. We then propose **Stee**rable **M**emory Agent, `SteeM`, a framework that allows users to dynamically regulate memory reliance, ranging from a fresh-start mode that promotes innovation to a high-fidelity mode that closely follows interaction history. Experiments across different scenarios demonstrate that our approach consistently outperforms conventional prompting and rigid memory masking strategies, yielding a more nuanced and effective control for personalized human-agent collaboration.

1 Introduction

Large language models are increasingly deployed as persistent agents capable of supporting users across extended timelines. To maintain continuity in these long-horizon interactions, systems are typically equipped with memory components that store user profiles, historical preferences, and past project states (Hu et al., 2025; Liu et al., 2025b). By retrieving and adding this context into the model's prompt, agents can achieve a high degree of personalization and consistency, effectively "picking up where they left off" rather than starting from scratch.

Current agent architectures predominantly treat memory retrieval as a static injection process. Once information is retrieved, the model often exhibits an *experience-following* tendency—i.e., retrieved records strongly steer the agent toward highly similar outputs (Xiong et al., 2025). However, in real-world scenarios, user requirements for memory usage are inherently dynamic (Cox and Ooi, 2022; Tversky and Simonson, 1993). For instance, a researcher may want an agent to act as a "project insider" that faithfully inherits prior decisions and constraints; yet, at other situations, they may require a "fresh-eyed reviewer" perspective that deliberately place less weight on legacy context to propose disruptive ideas. Existing systems struggle with this duality, often falling into *Memory Anchoring*: a state where the agent becomes overly constrained by its accumulated interaction history, failing to provide the clean-slate reasoning requested by the user (Laban et al., 2025; Lim et al., 2025; Dongre et al., 2025).

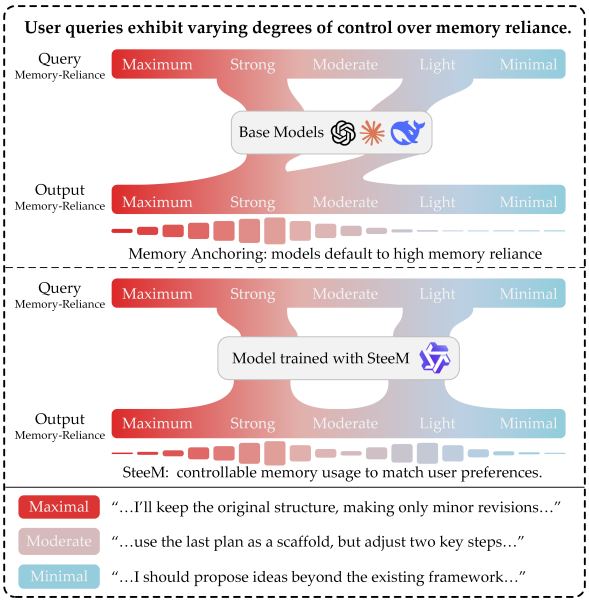

Figure 1: Illustration of *Memory Anchoring* and our solution `SteeM`, which steers model outputs to align with the user's memory-dependence preference.

The core of this problem is that current archi-

tectures lack a real-time mechanism for users to arbitrate memory dependence. Existing systems treat memory usage as a "black box" policy: once a memory is retrieved, its influence on the output is decided implicitly by the model's internal attention (Liu et al., 2025b; Zhang et al., 2025). Users are left with coarse, binary tools-either toggling memory "on or off" or manually masking items. Neither provides the ability to regulate behavioral dependence in real-time. Even when users explicitly prompt the model to "be creative" or "ignore previous drafts," LLMs often exhibit "memory leakage," where historical stylistic or ideological biases still bleed into the response. Consequently, the user — the only party with the context to know how much history is appropriate for the current task — is the one with the least control over it.

In this work, we propose a paradigm shift: the degree to which an agent leans on its long-term memory should be a user-controlled behavior dimension. We then introduce **Stee**rable **M**emory Agent, `SteeM`, a framework that enables users to dynamically control the degree to which model outputs rely on memory, ranging from a "bracketed" mode that prioritizes independent reasoning to a "high-fidelity" mode that strictly adheres to historical context. By treating memory dependence as a control axis, we empower users to navigate the trade-off between consistency and innovation based on their immediate, shifting needs. Specifically, we build a realistic dataset, simulating long-horizon human-agent interactions. We measure the memory dependence level of model outputs on this dataset, and develop `SteeM` that allows agents to follow a target dependence value across diverse scenarios. We demonstrate that our `SteeM` significantly outperforms prompt-based methods and memory masking, allowing users to achieve a far more precise balance between memory-awareness and reasoning independence across diverse long-horizon tasks.

2 Related Work

Alignment for Large Language Models Alignment is critical for improving user experience with LLM assistants, aiming to train models to better follow users' requests and generate outputs that better match human preferences (Ouyang et al., 2022). Common approaches include representation engineering (Liu et al., 2024), prompt optimization (Cheng et al., 2024; Wang et al., 2024a), SFT on demonstrations (Chung et al., 2024), direct preference optimization (DPO) from preference-pair data (Rafailov et al., 2023), and RL guided by a preference reward model (Ouyang et al., 2022; Schulman et al., 2017). Most prior work targets preferences over global response attributes, such as instruction following (Liu et al., 2025c) and HHH-style (helpful, honest, harmless) criteria (Bai et al., 2022). In contrast, our work focuses on a different preference axis: the user's intended degree of reliance on interaction memory, aligning generation to query-specific memory-dependence preferences.

Evaluating Personalization in Long-term Conversations Long-term conversation is a core application setting for LLM assistants, where personalization is critical to user experience (Zhang et al., 2025; Liu et al., 2025b). LoCoMo (Maharana et al., 2024) first evaluates LLMs on extremely long-term conversational histories and shows persistent failures in tracking long-range narratives and retrieving relevant context. PrefEval (Zhao et al., 2025a), PersonaMem-v1 (Jiang et al., 2025a) and PersonaMem-v2 (Jiang et al., 2025b) further introduce explicit or implicit user preferences and demonstrate that LLMs still struggle to produce preference-aligned responses over long interactions. However, two limitations remain in these studies: (1) they focus primarily on factual preference satisfaction, leaving preferences such as memory dependence underexplored despite its importance (Jones et al., 2025); (2) they implicitly assume that per-query preferences are consistent with prior interactions, although real preferences are intent-dependent and may naturally deviate from historical patterns (e.g., a usually rigorous user requesting an imaginative response) (Cox and Ooi, 2022; Tversky and Simonson, 1993). Our work aims to close these gaps by focusing on memory dependence preference and analyzing model performance under a dynamic preference setting.

Memory-Enhanced Personalized Agents To mitigate finite context windows and reduce interference from stale or irrelevant history in long-term conversations (Liu et al., 2025b; Wang et al., 2024b), recent agent systems introduce explicit retrievable memory modules that externalize interaction history into a persistent, continuously updated memory base (Hu et al., 2025; Zhong et al., 2023). By organizing and selectively retrieving from this memory base, the agent can construct a more query-relevant context for generation, improving long-horizon continuity and personalization (Liu et al.,

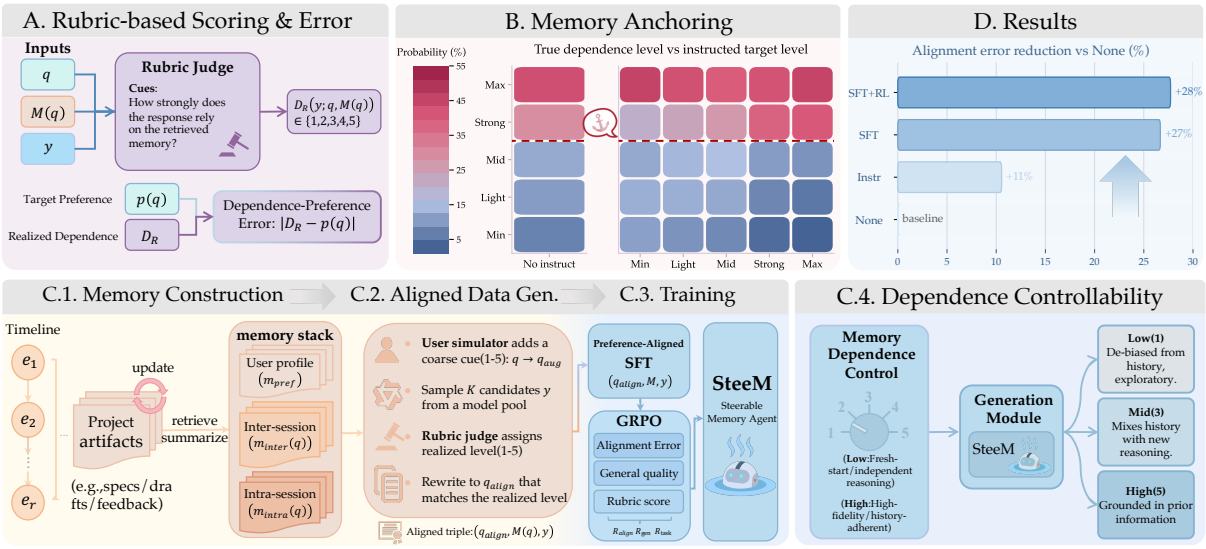

Figure 2: Overview of our approach and findings. (A) We use a rubric-based judge to score a response's memory dependence and compute the alignment error with targeted dependence. (B) We reveal *Memory Anchoring* in modern LLMs, where outputs default to high memory reliance despite low-dependence user intent. (C) We propose `SteeM`, built via a preference-aligned data generation pipeline followed by SFT and GRPO, enabling controllable memory usage. (D) `SteeM` achieves improved alignment to user-specified memory-dependence preferences.

2025b; Zhang et al., 2025). Representative systems include RMM (Tan et al., 2025), which combines multi-granularity summarization with retrospective retrieval refinement, LD-Agent (Li et al., 2025), which modularizes long-term personalization into independently tunable components, and O-Mem (Wang et al., 2025), which builds dynamic user profiles and performs hierarchical, user-centric retrieval. However, these systems provide limited transparency and user control over how strongly generation relies on retrieved memory (Xiong et al., 2025), despite evidence that users want mechanisms to regulate agents' access to memories (Jones et al., 2025). Our work analyzes memory's influence on outputs and proposes a framework for user-controllable memory dependence in generation.

3 Understanding *Memory Anchoring* with Realistic Synthetic Data

In this section, we first introduce a synthetic long-horizon pipeline for studying *Memory Anchoring* in agent generation and a rubric-based framework for measuring memory dependence.

3.1 Simulating Long-Horizon Interaction Histories

To study memory usage patterns of LLMs under long-horizon interactions, we simulate long-term projects as timelines of temporally ordered events and evolving project artifacts. On top of these, we subsequently instantiate task queries grounded in specific events and artifacts and derive query-specific memories from relevant subsets of the history, yielding a collection of $(q, M(q))$ instances that will later support our analysis of memory dependence and preference alignment.

Scenarios, Topics, Events, and Artifacts We instantiate two representative long-horizon **scenarios**, *Research* and *Tutoring*, covering common workflows in long-horizon human-agent interaction. We model each workflow as a timeline of scenario-specific **events** that drive progress (e.g., planning, experimentation, analysis for *Research*; teaching, practice, review for *Tutoring*) and a set of evolving **artifacts** that are produced and iteratively updated (e.g., experiment reports). For each scenario, we build a bank of 200 specific **topics** spanning diverse subjects by prompting Gemini-2.5-Pro and manually filtering for broad coverage and topical diversity. Each topic then serves as the seed for synthesizing a full project timeline with its associated events and artifacts. Table 3 lists all event and artifact types defined.

Iterative Timeline Synthesis Given a topic, we synthesize a project timeline as an ordered event sequence $\mathcal{T} = (e_1, \ldots, e_N)$ via an iterative generatevalidate loop. Each event e_t specifies an event

type, a brief description, prerequisite artifact types, and resulting artifact types that the event is expected to create or update. We maintain an artifact set $\mathcal{A}_t$ storing the latest version of each artifact. At each step t, we ask Gemini-2.5-Pro to propose the next event and corresponding artifacts conditioned on the topic, past events $(e_1, \ldots, et-1)$, and $\mathcal{A}_{t-1}$, yielding e_t and $\mathcal{A}_t$. After generation, we validate the proposal with (i) a prerequisite-type dependency check against $\mathcal{A}_{t-1}$ to ensure all required artifact types are available, and (ii) a global coherence check on e_t and $\mathcal{A}_t$ against the prior timeline to verify consistency. Invalid proposals are rejected and regenerated. We repeat this process until the timeline reaches a terminal state or a length limit.

Tasks and Queries We standardize tasks into four categories shared by both scenarios: *Plan & Design*, *Revise*, *Analyze & Critique*, and *Concept Explanation*. These tasks recur throughout long-horizon projects and can be answered either with minimal history or with strong reliance on prior context, enabling controlled evaluation of memory dependence. We instantiate queries by grounding tasks on specific events and artifacts in the timeline. Each query is constructed from a triplet $q = \langle e_t, \text{task}, \text{target} \rangle$, where e_t denotes the triggering event, task specifies the task type, and target is the artifact to be operated on. Given the post-event artifact set $\mathcal{A}_t$, we sample $(\text{task}, \text{target})$ and generate the natural-language query using a task-specific template.

Query-Specific Memory Construction For each query q triggered at event e_t, we construct a query-specific memory:

$$M(q) = \{m_{\text{prof}},\, m_{\text{inter}}(q),\, m_{\text{intra}}(q)\}, \quad (1)$$

where m_{prof} encodes long-term user goals and preferences, $m_{\text{inter}}(q)$ summarizes relevant cross-session interactions, and $m_{\text{intra}}(q)$ summarizes the recent intra-session history. These components are derived from the synthetic timeline and artifacts by selecting query-relevant items and rewriting them into concise natural-language summaries. The resulting memory $M(q)$ serves as the simulated retrieved context for the specific query q.

Dataset Statistics The pipeline yields a diverse and realistic synthetic dataset with over 7,000 events, 7,000 artifacts, and 10,000+ $(q, M(q))$ pairs. Detailed statistics are presented in Table 2 and Figure 8. We reserve a held-out test set of 1000 $(q, M(q))$ pairs with uniform coverage across scenarios and tasks for later use.

3.2 Formulating Memory-Dependence Preference

Building on the synthetic $(q, M(q))$, we now formalize memory dependence and user preference over it. Given a user query q and its query-specific memory $M(q)$, the model parameterized by θ generates a response $y \sim \pi_\theta(\cdot \mid q, M(q))$. To quantify the reliance of a response on $M(q)$ beyond a binary use or not judgment, we introduce a rubric-based **memory-dependence metric**:

$$D^q_{\mathcal{R}}(y) \;\triangleq\; D_{\mathcal{R}}\big(y;\, q, M(q)\big) \;\in \{1,2,3,4,5\}, \quad (2)$$

where $\mathcal{R}$ is a set of human-aligned rubrics spanning memory-agnostic to strongly memory-grounded behaviors. We refer to $D^q_{\mathcal{R}}(y)$ as the **memory-dependence score (MD-Score)** of y, where larger values indicate stronger reliance on $M(q)$. $D_{\mathcal{R}}(\cdot)$ is implemented as an LLM-as-a-judge evaluator that assigns scores on this 1–5 scale using $\mathcal{R}$.

Memory-Dependence Preference. Building on the rubric set $\mathcal{R}$, we formalize the query-specific target degree of reliance on $M(q)$ in generation as **memory-dependence preference (MD-Pref)**, denoted by $p(q) \in \{1,2,3,4,5\}$ on the same $\mathcal{R}$-defined scale used by $D_{\mathcal{R}}(\cdot)$. With $(q, M(q), y)$ and $p(q)$, we define the **alignment error of MD-Pref** $\delta_{\text{align}}(q, M(q), y)$:

$$\delta_{\text{align}}(q, M(q), y) = \big|D_{\mathcal{R}}\big(y; q, M(q)\big) - p(q)\big|, \quad (3)$$

which measures how closely y matches the target dependence level $p(q)$ specified by the user.

3.3 *Memory Anchoring* in Agent Generation

We first run a human study to verify that the rubric-based MD-Score matches human judgments of memory reliance, and then use it to characterize agent behavior when memory is available.

Pairwise Validity of MD-Score With the test set obtained from Section 3.1, we sample multiple responses per query using different prompting settings and models, and compute their MD-Scores $D_{\mathcal{R}}$. For each query q, we randomly select two responses with different MD-Scores to form a pair $(y^{(1)}, y^{(2)})$. Human annotators are shown the same $(q, M(q))$ and asked to judge which response relies more on the provided memory. We treat $\text{sign}\big(D^q_{\mathcal{R}}(y^{(1)}) - D^q_{\mathcal{R}}(y^{(2)})\big)$ as the metric's estimated pairwise ranking, and report its agreement

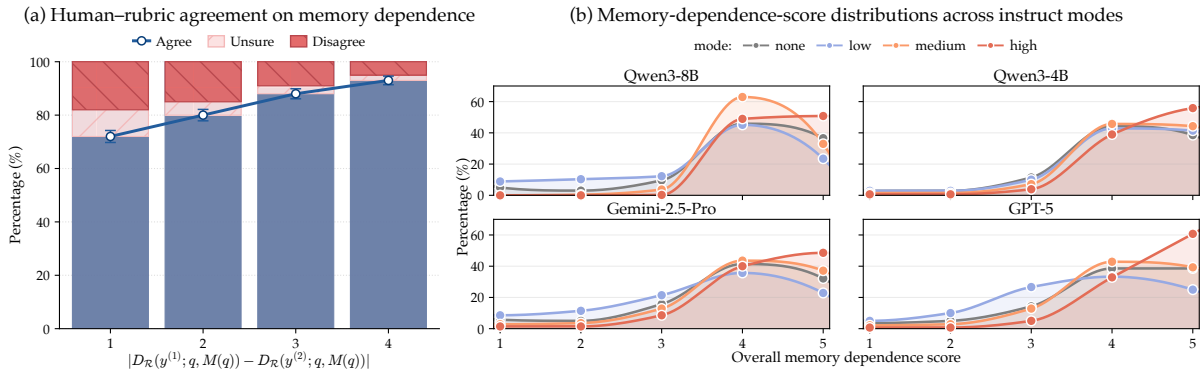

Figure 3: Human–judge agreement on memory-dependence comparisons (left) and memory-dependence score distributions across models and dependence prompts (right).

rate and rank correlation with human judgments (Figure 3, left). We observe strong consistency, especially when the score gap $|D_{\mathcal{R}}^{q}(y^{(1)}) - D_{\mathcal{R}}^{q}(y^{(2)})|$ is large, supporting $D_{\mathcal{R}}$ as a proxy for memory dependence. Annotation details are in Appendix G.

Prompting-based Control and *Memory Anchoring* We examine whether natural-language prompting alone can regulate memory reliance on modern LLMs, including Qwen3-4B/8B, Gemini-2.5-Pro, and GPT-5 (Yang et al., 2025; Comanici et al., 2025; OpenAI, 2025). We evaluate four dependence modes: NONE (no additional instruction) and three rubric-aligned prompts with targeted levels $\ell \in \{\text{LOW}, \text{MEDIUM}, \text{HIGH}\}$, corresponding to rubric levels $\{1, 3, 5\}$ in $\mathcal{R}$. We prepend a mode-specific instruction that specifies the desired dependence level ℓ or NONE to the original query q. Full prompts are provided in Appendix F. For each setting, we perform inference on the test set and compute the empirical distribution of $D_{\mathcal{R}}^{q}(y)$ over queries (Figure 3, right). Across all models, the distributions concentrate on high dependence (scores 4-5), and switching the prompt from LOW to HIGH yields only marginal shifts. This suggests that once memory is available, LLMs default to strong memory reliance, and prompt-only dependence instructions have limited control over the realized level. We refer to this persistent high-dependence generation behavior despite explicit user instructions as *Memory Anchoring*, motivating more explicit mechanisms for regulating memory usage.

4 Method

4.1 Problem Formulation

Given a query q and its constructed memory $M(q)$, our goal is to generate a response that matches the users query-specific memory-dependence preference $p(q)$. Formally, with $y \sim \pi_\theta(\cdot \mid q, M(q))$, we optimize parameters θ to minimize the **alignment error of dependence preference** defined in Equation (3):

$$\min_{\theta} \ \delta_{\text{align}}(q, M(q), y) \tag{4}$$

In the following, we pursue this objective via preference-aware supervised fine-tuning and reinforcement learning, encouraging the model response to match $p(q)$ while preserving task quality.

4.2 Memory-Dependence Aligned Supervised Fine-Tuning

As analyzed in Section 3.3, current models suffer from *memory anchoring*, tending to produce heavily memory-reliant responses even when instructed with low memory-dependence preference. This makes it difficult to obtain ideal training data with low δ_{align} via a naive sample-and-filter strategy. To address this, we introduce an efficient pipeline that automatically generates high-quality training data.

Preference-Aligned Data Generation To ensure diversity of training data across different dependence levels, we first augment each preference-agnostic original query q with a target memory-dependence preference $p_{\text{aug}} \in \{1, 2, 3, 4, 5\}$. To elicit natural preference expressions, we employ a user simulator powered by Gemini-2.5-Pro. We provide the user simulator with $(q, M(q))$ and a target dependence level p_{aug} described only coarsely (without revealing the full rubric set $\mathcal{R}$), and ask it to rewrite q into a preference-indicative query q_{aug} that implicitly conveys the the semantics of p_{aug}. Given each $(q_{\text{aug}}, M(q))$ pair, we then sample 4 candidate responses $y \sim \pi(\cdot \mid q_{\text{aug}}, M(q))$

from a pool of models (Qwen3-8B, Qwen3-14B), yielding diverse outputs under preference-guided prompting. For each candidate y, we compute $D_{\mathcal{R}}\big(y; q, M(q)\big)$ with respect to the original query q to obtain its realized dependence level. Although these responses are generated with an augmented query q_{aug}, they do not necessarily match the target dependence preference p_{aug}, as observed in Section 3.3. Therefore, we invoke the user simulator once more to rewrite the original query q into an aligned variant q_{align} whose implicit preference matches the realized dependence score of the corresponding y, such that $p(q_{\text{align}}) = D_{\mathcal{R}}\big(y; q, M(q)\big)$. Substituting the preference-agnostic q with q_{align}, we finally obtain preference-aligned training triples $(q_{\text{align}}, M(q), y)$.

Quality-Preserving Filtering Preference alignment alone may admit low-quality generations, which is unacceptable for good user experience. To preserve response quality, we additionally score each retained candidate using (1) task-oriented general rubrics and (2) a reward model. We keep only the highest-scoring subset for an original query q, yielding a final 7000 SFT set $\mathcal{D}_{\text{SFT}} = \{(q_{\text{align}}, M(q), y)\}$ that is both aligned and high-quality.

Supervised Fine-Tuning We fine-tune Qwen3-4B and Qwen3-8B (Yang et al., 2025) on $\mathcal{D}_{\text{SFT}}$ with the standard token-level cross-entropy objective.

4.3 δ_{align}-Guided Reinforcement Learning

After SFT, we further optimize the policy with RL on the preference-indicative inputs $(q_{\text{align}}, M(q))$. We adopt GRPO with a carefully designed reward that jointly promotes memory-dependence alignment and task quality.

Reward Design Our reward signal R comprises three components. First, we use the alignment error $\delta_{\text{align}}(q_{\text{align}}, M(y), y)$ as a direct supervision signal for memory-dependence preference satisfaction. Since a lower δ_{align} indicates better alignment, we convert it into an **alignment reward**:

$$\begin{aligned} R_{\text{align}}(q_{\text{align}}, y) &= -\,\delta_{\text{align}}(q_{\text{align}}, M(q), y) \\ &= -\big|D_{\mathcal{R}}\big(y; q_{\text{align}}, M(q)\big) - p(q_{\text{align}})\big| \end{aligned} \tag{5}$$

Second, to preserve task-related correctness and usefulness, we assign each response a rubric-based **task reward** $R_{\text{task}}(q_{\text{align}}, y)$ on a 1-5 scale, where higher is better. Third, we incorporate **general reward** $R_{\text{general}}(q_{\text{align}}, y)$ scored by a reward model to guarantee the general quality of the responses.

We aggregate these signals to form the final reward:

$$R = R_{\text{align}} + R_{\text{task}} + R_{\text{general}}. \tag{6}$$

RL Objective We optimize π_θ with GRPO, maximizing a group-based clipped objective:

$$\max_{\theta}\ \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\min\Big(\rho^{(k)}\hat{A}^{(k)}, \text{clip}(\rho^{(k)}, 1-\epsilon, 1+\epsilon)\hat{A}^{(k)}\Big)\right],$$
$$\rho^{(k)} \triangleq \frac{\pi_\theta(y^{(k)} \mid q_{\text{align}}, M(q))}{\pi_{\theta_{\text{old}}}(y^{(k)} \mid q_{\text{align}}, M(q))}, \quad \hat{A}^{(k)} \triangleq R^{(k)} - \frac{1}{K}\sum_{j=1}^{K} R^{(j)} \tag{7}$$

RL Data. We select 2000 samples that do not overlap with the SFT dataset for RL. We uniformly assign each original sample a target preference $p(q)$ and then augment it into preference-indicative queries using the same pipeline described in Section 4.2.

5 Experiments

5.1 Main Results

We examine model performance in terms of (i) alignment with the target memory-dependence level, (ii) response quality, and (iii) generalizability to queries about unseen subjects.

Baselines For a fair comparison, we consider two baselines: *None*, which measures the base models performance on preference-indicative queries, and *Rubric Instruct*, which evaluates the base model when explicitly prompted with the rubrics corresponding to the target dependence level.

Test Data We use the test set produced in Section 3.1. Similarly, we also augment them to be preference-indicative as described in Section 4.2.

5.1.1 Steering Outputs Toward User-Preferred Memory Dependence

Overview of Alignment Results We evaluate whether `SteeM` can steer generations toward the memory-dependence preference implicitly expressed in each query. Across the *Research* and *Tutoring* scenarios, we measure the dependence-preference alignment error δ_{align} on four shared tasks. As shown in Table 1, `SteeM` consistently achieves substantially lower δ_{align} than the baseline across all scenarios and tasks. This indicates that

Method	RESEARCH				TUTORING				Avg. ↓
	Plan & Design	*Revise*	*Analyze & Critique*	*Concept Explanation*	*Plan & Design*	*Revise*	*Analyze & Critique*	*Concept Explanation*	
proprietary Models									
Gemini-2.5-Pro	1.34	1.61	1.52	1.13	1.43	1.64	1.50	1.36	1.44
GPT-5	1.28	1.56	1.50	1.02	1.51	1.59	1.50	1.22	1.40
Qwen3-4B									
None	1.81 ↓0.00	1.76 ↓0.00	1.58 ↓0.00	1.20 ↓0.00	1.68 ↓0.00	1.77 ↓0.00	1.65 ↓0.00	1.23 ↓0.00	1.59 ↓0.00
Rubric Instruct	1.46 ↓0.35	1.69 ↓0.07	1.49 ↓0.09	1.03 ↓0.17	1.50 ↓0.18	1.74 ↓0.03	1.58 ↓0.07	1.04 ↓0.19	1.44 ↓0.14
SteeM (SFT)	1.14 ↓0.67	1.54 ↓0.22	1.12 ↓0.46	0.95 ↓0.25	**1.32** ↓0.36	1.51 ↓0.26	1.41 ↓0.24	0.91 ↓0.32	1.24 ↓0.35
SteeM (SFT+RL)	**1.01** ↓0.80	**1.53** ↓0.23	**1.11** ↓0.47	**0.87** ↓0.33	**1.32** ↓0.36	**1.46** ↓0.31	**1.38** ↓0.27	**0.86** ↓0.37	**1.19** ↓0.39
Qwen3-8B									
None	1.69 ↓0.00	1.76 ↓0.00	1.54 ↓0.00	1.12 ↓0.00	1.70 ↓0.00	1.75 ↓0.00	1.61 ↓0.00	1.35 ↓0.00	1.57 ↓0.00
Rubric Instruct	1.31 ↓0.38	1.57 ↓0.19	1.44 ↓0.10	1.02 ↓0.10	1.65 ↓0.05	1.72 ↓0.03	1.49 ↓0.12	1.00 ↓0.35	1.40 ↓0.17
SteeM (SFT)	1.02 ↓0.67	1.35 ↓0.41	**1.07** ↓0.47	0.88 ↓0.24	**1.25** ↓0.45	1.48 ↓0.27	1.26 ↓0.35	0.87 ↓0.48	1.15 ↓0.42
SteeM (SFT+RL)	**0.99** ↓0.70	**1.33** ↓0.43	1.09 ↓0.45	**0.83** ↓0.29	1.28 ↓0.42	**1.43** ↓0.32	**1.25** ↓0.36	**0.85** ↓0.50	**1.13** ↓0.43

Table 1: δ_{align} across scenarios and tasks. Lower is better. Our SteeM achieves the lowest alignment error on memory-dependence preferences.

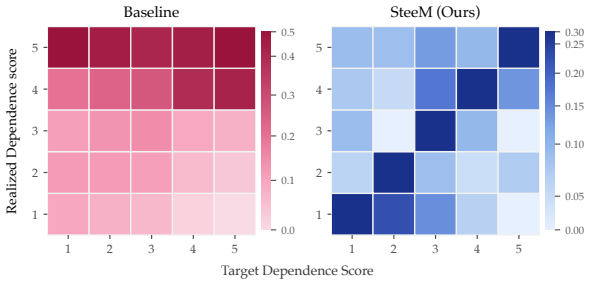

Figure 4: Realized dependence levels $D^q_{\mathcal{R}}(y)$ conditioned on the target preference $p(q)$. Columns are target levels and rows are realized levels (column-normalized). SteeM concentrates more mass near the diagonal than the baseline.

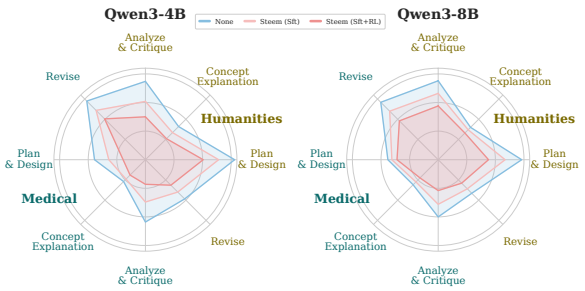

Figure 5: Radar plots of the **alignment error** on unseen subjects settings (Medical and Humanities). Curves closer to the center indicate better alignment.

SteeM produces responses whose realized memory dependence more closely matches the user-preferred dependence level implied by the query, enabling a better control of memory usage.

Distribution of Realized vs. Target Dependence Levels To better understand how alignment behaves across dependence levels, we sample 100 queries per level and visualize the distribution of realized levels conditioned on the target $p(q)$ as a confusion-matrix heatmap. Figure 4 plots the confusion matrices between target levels $p(q)$ and realized levels $D_{\mathcal{R}}(y; q, M(q))$. Compared to the baseline, which exhibits a strong memory-anchoring bias with most mass concentrated at high realized levels (4–5) regardless of the target, SteeM significantly shifts the distribution toward the diagonal, indicating substantially improved alignment to the intended dependence level.

Generalizing to Unseen Subjects To assess the generalizability of SteeM, we further evaluate it on queries from previously unseen subjects in the *Research* scenario: Medical and Humanities. Figure 5 shows that SteeM learns preference-following behavior from the training data and transfers it to new subjects, with the RL-enhanced variant exhibiting stronger generalization than SFT alone (a bigger gap compared with the results in Table 1).

5.1.2 Preserving Response Quality

A key concern in steering outputs toward memory-dependence preferences is whether alignment comes at the cost of utility. To verify this, we evaluate model generations using an overall reward score computed by Skywork-Reward-V2-Llama-3.1-8B (Liu et al., 2025a), a strong and widely adopted reward model. Results in Figure 6 show that SteeM maintains response quality comparable to the baseline, and even yields slightly higher

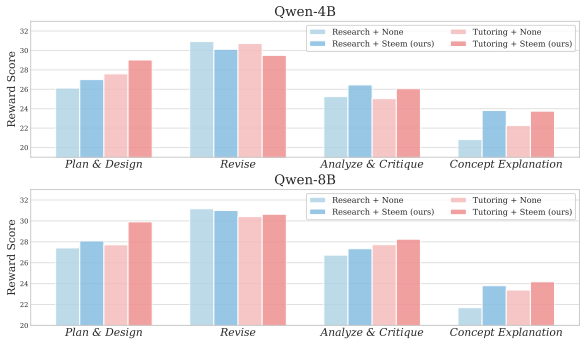

Figure 6: Comparison of response quality across models, scenarios, tasks.

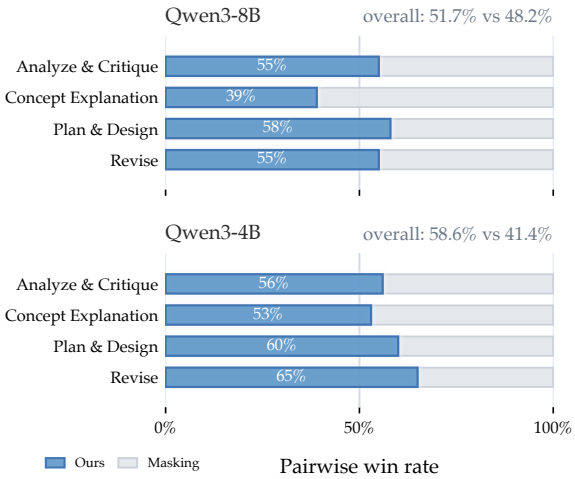

Figure 7: SteeM vs. memory masking. Task-wise pairwise win rates on Qwen3-8B and Qwen3-4B.

scores in several cases. We further report reward scores on a general benchmark, AlpacaEval, in Table 4. The results suggest that SteeM improves preference alignment while introducing only a minimal impact on general response quality.

5.2 Natural Expression vs. Predefined-Tag

A straightforward way to control memory dependence is to train on queries augmented with predefined tags that explicitly specify the target dependence level. To compare this with the natural preference expressions used in SteeM, we train a tag-conditioned variant using the same data pipeline and optimization recipe, but replacing implicit preference cues with five predefined tags (from Minimal to Maximal). Tables 4 and 6 show that tag-conditioned training achieves slightly better alignment than SteeM, but significantly degrades general performance on AlpacaEval.

5.3 Comparison with Straightforward Binary Memory Masking

A straightforward baseline for controlling memory dependence is *memory masking*, which directly masks a portion of memory according to the target preference $p(q)$. We implement this by using an LLM-based user simulator to select a subset of memories based on the preference before generation. We compare this baseline with SteeM via pairwise LLM-as-a-judge evaluation, asking which response better matches $p(q)$ and completes the task. As shown in Figure 7, SteeM is competitive with masking and yields a consistent win-rate advantage, highlighting a key limitation of masking: it changes what information is available, but cannot reliably regulate how strongly the model relies on memory. Moreover, masking may drop critical constraints or facts and places a heavy selection burden on users in long, information-dense histories.

5.4 Case Study

Table 5 qualitatively illustrates our main contribution: models often over-use the given memory, while our SteeM can steer generation toward the user-intended degree of memory reliance. The case requests new ideas with low memory dependence to refine a PROJECT_METHOD under a topic of curriculum-learning recipe. The baseline response largely follows the historical pipeline (blue) with only minor add-ons, reflecting memory anchoring despite the user instruction. In contrast, SteeM introduces more substantial departures (red), such as adaptive sampling and progress-triggered transitions. Overall, SteeM better matches the user's low-memory intent and reduces unintended memory-following.

6 Conclusion

We study an important yet underexplored user preference in long-horizon interactions: how much an agent should rely on historical memory. We build a realistic dataset simulating long-horizon interactions and identify *memory anchoring*, where models default to high memory reliance despite user intent. To address this, we propose SteeM, trained with preference-aligned SFT and RL, which achieves substantially better preference alignment. It transfers well beyond our controlled long-horizon setting with minimal impact on general performance, and outperforms direct memory masking in pairwise comparisons. We hope our study offers an initial step toward practical, user-controllable memory reliance for personalized agents.

Limitations

While we make a concerted effort to mimic realistic long-horizon projects and believe it is enough to serve as a useful testbed for studying Memory Anchoring, it may still differ from real human interactions. We model memory-dependence preference on a 15 ordinal, whereas real users may express richer and more nuanced constraints. Future work could extend this formulation to a finer-grained or even continuous spectrum. In addition, our current setup covers only two scenarios, *Research* and *Tutoring*. Extending the data and evaluation to broader application settings and more diverse task distributions remains an important direction.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2024. Black-box prompt optimization: Aligning large language models without model training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3201–3219.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Samuel Rhys Cox and Wei Tsang Ooi. 2022. Does chatbot language formality affect users self-disclosure? In *Proceedings of the 4th Conference on Conversational User Interfaces*, CUI '22, New York, NY, USA. Association for Computing Machinery.

Vardhan Dongre, Ryan A. Rossi, Viet Dac Lai, David Seunghyun Yoon, Dilek Hakkani-Tür, and Trung Bui. 2025. Drift no more? context equilibria in multi-turn LLM interactions. *Preprint*, arXiv:2510.07777.

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.

Bowen Jiang, Zhuoqun Hao, Young Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo Jose Taylor, and Dan Roth. 2025a. Know me, respond to me: Benchmarking LLMs for dynamic user profiling and personalized responses at scale. In *Second Conference on Language Modeling*.

Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh Rao Vijjini, Jiashu He, Hanchao Yu, Radha Poovendran, Gregory Wornell, Lyle Ungar, Dan Roth, Sihao Chen, and Camillo Jose Taylor. 2025b. Personamem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory. *Preprint*, arXiv:2512.06688.

Brennan Jones, Kelsey Stemmler, Emily Su, Young-Ho Kim, and Anastasia Kuzminykh. 2025. Users' expectations and practices with agent memory. CHI EA '25, New York, NY, USA. Association for Computing Machinery.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. Llms get lost in multi-turn conversation. *Preprint*, arXiv:2505.06120.

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2025. Hello again! LLM-powered personalized agent for long-term dialogue. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5259–5276, Albuquerque, New Mexico. Association for Computational Linguistics.

Seungseop Lim, Gibaeg Kim, Wooseok Han, Jean Seo, Hyunkyung Lee, Jaehyo Yoo, and Eunho Yang. 2025. Format inertia: A failure mechanism of LLMs in medical pre-consultation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1437–1450, Suzhou (China). Association for Computational Linguistics.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. 2025a. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*.

Jiahong Liu, Zexuan Qiu, Zhongyang Li, Quanyu Dai, Wenhao Yu, Jieming Zhu, Minda Hu, Menglin Yang, Tat-Seng Chua, and Irwin King. 2025b. A survey of personalized large language models: Progress and future directions. *arXiv preprint arXiv:2502.11528*.

Wenhao Liu, Zhengkang Guo, Mingchen Xie, Jingwen Xu, Zisu Huang, Muzhao Tian, Jianhan Xu, Muling

Wu, Xiaohua Wang, Changze Lv, and 1 others. 2025c. Recast: Strengthening llms' complex instruction following with constraint-verifiable data. *arXiv preprint arXiv:2505.19030*.

Wenhao Liu, Xiaohua Wang, Muling Wu, Tianlong Li, Changze Lv, Zixuan Ling, Zhu JianHao, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2024. Aligning large language models with human preferences through representation engineering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10619–10638, Bangkok, Thailand. Association for Computational Linguistics.

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

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A Smith, Hannaneh Hajishirzi, and Nathan Lambert. 2025. Rewardbench 2: Advancing reward model evaluation. *arXiv preprint arXiv:2506.01937*.

OpenAI. 2025. GPT-5 System Card. https://cdn.openai.com/gpt-5-system-card.pdf. Accessed: 2026-01-05.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Rajan Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. 2025. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8416–8439, Vienna, Austria. Association for Computational Linguistics.

Amos Tversky and Itamar Simonson. 1993. Context-dependent preferences. *Management Science*, 39(10):1179–1189.

Piaohong Wang, Motong Tian, Jiaxian Li, Yuan Liang, Yuqing Wang, Qianben Chen, Tiannan Wang, Zhicong Lu, Jiawei Ma, Yuchen Eleanor Jiang, and Wangchunshu Zhou. 2025. O-mem: Omni memory system for personalized, long horizon, self-evolving agents. *Preprint*, arXiv:2511.13593.

Xiaohua Wang, Zisu Huang, Feiran Zhang, Zhibo Xu, Cenyuan Zhang, Qi Qian, Xiaoqing Zheng, and Xuanjing Huang. 2024a. Enhancing the capability and robustness of large language models through reinforcement learning-driven query refinement. *arXiv preprint arXiv:2407.01461*.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, Ruicheng Yin, Changze Lv, Xiaoqing Zheng, and Xuanjing Huang. 2024b. Searching for best practices in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17716–17736, Miami, Florida, USA. Association for Computational Linguistics.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. 2025. How memory management impacts llm agents: An empirical study of experience-following behavior. *Preprint*, arXiv:2505.16067.

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

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, and 2 others. 2025. Personalization of large language models: A survey. *Transactions on Machine Learning Research*. Survey Certification.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. 2025a. Do LLMs recognize your preferences? evaluating personalized preference following in LLMs. In *The Thirteenth International Conference on Learning Representations*.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, and 1 others. 2025b. Swift: a scalable lightweight infrastructure for fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 29733–29735.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2023. Memorybank: Enhancing large language models with long-term memory. *Preprint*, arXiv:2305.10250.

Category	Research	Tutoring	Total
Interaction-history Statistics			
Timelines	200	200	400
Events	3534	4005	7539
Artifacts	3850	3895	7745
Task Statistics			
Plan & Design	1214	2194	3408
Revise	1823	2211	4034
Analyze & Critique	1298	1474	2772
Concept Explanation	607	720	1327

Table 2: Statistics of our synthetic dataset across scenarios.

Scenario	Category	Types

Research	Event	`proposal`
		`method_design`
		`pilot_experiments`
		`main_experiments`
		`analysis`
		`writing`

	Artifact	`research_plan`
		`research_goals`
		`experiment_results`
		`method`
		`paper_paragraph`

Tutoring	Event	`objective_clarification`
		`plan_milestones`
		`lesson`
		`practice`
		`review`
		`materials_revision`

	Artifact	`learning_objectives`
		`study_plan`
		`teaching_notes`
		`practice_record`
		`feedback_summary`

Table 3: Scenario-specific event and artifact type definitions.

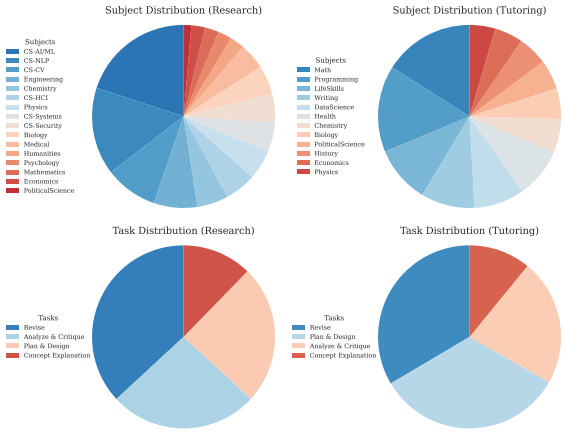

Figure 8: Distribution of subjects and tasks in our simulated real-world interaction dataset.

A Dataset Details

Scenarios and Topics We instantiate two representative long-term project scenarios: *Research* and *Tutoring*. They cover two common forms of sustained human-agent collaboration: (1) open-ended research projects that evolve through planning, experimentation, analysis, and writing; and (2) tutoring projects that proceed via goal setting, lesson delivery, practice, and review. Each scenario is treated as a project "container" within which the agent and user interact over an extended timeline. For each scenario, we first define a set of coarse-grained subjects. We then build a bank of 200 topics per scenario by prompting Gemini-2.5-Pro to propose candidate project themes and manually filtering them to ensure broad coverage and topical diversity.

Events and Artifacts We predefine scenario-specific *event types* and *artifact types* to reflect the core structure of each long-term scenario. Event types represent key milestones that mark meaningful progress in the project trajectory, while artifact types correspond to essential intermediate products that are produced and iteratively updated throughout the process. Table 3 lists all event and artifact types used in our two scenarios.

Iterative Timeline Synthesis Given a topic, we synthesize a project timeline as an ordered sequence of events $\mathcal{T} = (e_1, e_2, \ldots, e_N)$ using an iterative generation-validation protocol. Each event e_t is a structured record with an event type, a natural-language description, and lists of prerequisite and produced artifact types. We maintain a running artifact set $\mathcal{A}_t$ that stores the latest version of each artifact. At step t, Gemini-2.5-Pro proposes a candidate next event e_t conditioned on the topic, the past events $(e_1, \ldots, e_{t-1})$, and $\mathcal{A}_{t-1}$. We then (i) perform a symbolic dependency check to ensure that all prerequisite artifact types are present in $\mathcal{A}_{t-1}$, rejecting and regenerating events that violate these constraints, and (ii) update $\mathcal{A}_t$ with the produced artifact types and ask Gemini-2.5-Pro to assess the global coherence of the updated timeline (e.g., logical consistency and compatibility with earlier decisions). We repeat this process until the project reaches a natural terminal state or a predefined maximum length. This dependency-constrained, multi-step protocol yields realistic project trajectories in which progress arises from coherent updates to existing artifacts and occa-

sional backtracking or refinement (e.g., revising goals or rerunning experiments). Generation and validation prompts are provided in the appendix.

Tasks and Queries To make model behavior on specific queries comparable, we standardize the task interface into four generic categories shared by both scenarios: *Plan & Design*, *Revise*, *Analyze & Critique*, and *Concept Explanation*. These tasks (i) cover common information-seeking needs that naturally arise at multiple stages of long-term projects, and (ii) admit both history-agnostic and strongly history-dependent solutions for the same query, which is crucial for probing controllable memory usage without conflating it with changes in task form. We instantiate queries by attaching these tasks to specific events and artifacts on the timeline. Formally, each query is a triplet $q = \langle e_t, \text{task}, \text{target} \rangle$, where e_t is the associated event, task is one of the four categories above, and target is an artifact to operate on (e.g., a draft section, an experiment report, or a homework solution). We treat q as a natural user question issued immediately after e_t completes. Concretely, given the post-event artifact set $\mathcal{A}_t$, we select a feasible task category, sample a suitable target artifact, and generate the query text by filling a task-specific template with the topic and target information.

Query-Specific Memory Construction For each query q anchored at event e_t, we construct a query-specific memory $M(q)$. We decompose it into three components:

$$M(q) = \{m_{\text{prof}}, m_{\text{inter}}(q), m_{\text{intra}}(q)\}. \quad (8)$$

Here m_{prof} is a user profile capturing long-term goals and preferences, $m_{\text{inter}}(q)$ summarizes relevant cross-session or cross-topic interactions, and $m_{\text{intra}}(q)$ summarizes the recent within-session history around e_t. All three components are derived from the synthetic timelines and artifacts by selecting relevant events and artifacts for q and rewriting them as concise natural-language summaries. The resulting memory $M(q)$, together with q, forms the retrieved context for the model and provides a handle to vary how much history is exposed when analyzing and controlling memory dependence.

Dataset Statistics The above meticulous data synthesis pipeline finally produces a diverse and realistic synthetic dataset, whose statistics are presented in Table 2 and Figure 8.

Model	Method	AlpacaEval score
Qwen-4B	None	8.85
	Tag Cue	
	Tag-cued SFT-only	8.33
	Tag-cued RL	8.45
	STEEM	
	STEEM SFT-only	8.59
	STEEM RL	8.73
Qwen-8B	None	10.49
	Tag Cue	
	Tag-cued SFT-only	10.02
	Tag-cued RL	10.14
	STEEM	
	STEEM SFT-only	10.12
	STEEM RL	10.43

Table 4: AlpacaEval scores across methods and models. We report the mean reward scores.

B Training Details

Supervised Fine-Tuning We perform SFT using the MS-SWIFT (Zhao et al., 2025b) training framework with a global batch size of 64, a learning rate of 1×10^{-5}, and 3 training epochs.

Reinforcement Learning We perform GRPO (Shao et al., 2024) using the EASYR1 framework with a rollout batch size of 32, an update batch size of 8, a learning rate of 5×10^{-6}, a maximum sequence length of 6144 tokens, and 8 rollouts per prompt.

Training Data After the synthesis pipeline described in Section 4.2, we finally gain 7000 aligned SFT samples and 2000 RL samples.

C Response Quality

We report all AlpacaEval results in Table 4. For scoring, we use Skywork-Reward-V2-Llama-3.1-8B (Liu et al., 2025a) as the reward model, which is a strong open-source RM and performs competitively on RewardBench-2 (Malik et al., 2025).

D Case Study

Table 5 qualitatively illustrates our main contribution: models often over-use the given memory, while our `SteeM` can steer generation toward the user-intended degree of memory reliance. The case was requested of new ideas with low memory dependence to refine a PROJECT_METHOD under a topic of curriculum-learning recipe. The

Historical Artifacts: PROJECT_METHOD (Topic: Curriculum learning for foundation models)	
Difficulty Metric: Perplexity assigned by a pre-trained model is used to score each document (lower perplexity → lower difficulty). **Data Organization:** Sort by difficulty and partition into 10 equal buckets (B1 easiest → B10 hardest). **Pacing Function:** Linear step-based pacing: start from B1; every 10% of total steps add the next hardest bucket; after 90% train on the full dataset.	
Query	
Improve the project method. Ignore the historical framework, I need some new idea and insights.	
Responses	
Baseline	**SteeM (ours)**
- **Difficulty signal**: ..... use DistilGPT-2 perplexity as the primary difficulty score; add a light normalization step (e.g., length-normalized perplexity) to reduce bias ..... - **Bucketing**: sort examples by difficulty and split them into 10 equal buckets (B1 easiest B10 hardest) ..... - **Pacing**: adopt a linear, step-based schedulestart from B1, add one bucket every 10% of total steps, then mix over all buckets after 90% ..... - **Sanity check**: log bucket-level loss/perplexity to verify the bucket ordering remains stable over training	- **Curriculum Signal**: ..... Use a two-factor signal: LM perplexity (DistilGPT-2) + a coverage indicator to discourage over-sampling near-duplicates ..... - **Sampling Policy**: Adopt mixture-of-difficulties sampler: at step t, sample from a distribution P_t over difficulty quantiles and update P_t ..... - **Progress Controller**: Use progress-triggered curriculum transitions rather than fixed strategy ..... If instability occurs, automatically narrow P_t ..... - **Curriculum Diagnostics**: Track coverage entropy, quantile-wise learning progress and instability flags (loss spikes / grad-norm outliers)..... - **Compatibility Note**: Retain the original bucketed curriculum structure for comparability .....

Table 5: Case study on PROJECT_METHOD refinement under a user request for a **low-memory-dependence response**. Blue highlights mark content that follows historical artifacts (memory-following), while red highlights indicate newly introduced ideas (creative additions). Compared to the baseline, SteeM produces a more creative, less memory-dependent update that better matches the user's intent.

baseline response largely follows the historical pipeline (blue) with only minor add-ons, reflecting memory anchoring despite the user instruction. In contrast, SteeM introduces more substantial departures (red), such adaptive sampling and progress-triggered transitions. Overall, SteeM better matches the users low-memory intent and reduces unintended memory-following.

E Details for Natural Expression vs. Predefined-Tag Comparison

We present the detailed comparison between tag-cued training and our SteeM in Table 4 and Table 6.

F Memory-Dependence Rubrics

We provide the full memory-dependence judging rubric $\mathcal{R}$ used to assign the integer MD-Score $D_{\mathcal{R}}$ in our experiments. The complete rubric (including scale definitions and dimension-wise guidance) is shown in Table 7.

G Human Annotation Protocol

We provide the annotation protocol used in human-correlation analysis of Section 3.3. We annotate 1000 pairwise comparison instances. Each instance contains the same $(q, M(q))$ and two candidate responses, and the annotator selects which response relies *more* on the provided memory; the exact annotation prompt is shown in Figure 11. These instances are randomly partitioned into 10 shards and assigned to 10 volunteer annotators (100 instances per annotator). Each judgment requires reading the shared context and comparing two responses; we estimate an average of ∼45 seconds per instance, yielding an estimated workload of ∼75 minutes per annotator. All annotators participated on an interest-driven, voluntary basis. The resulting agreement and rank correlation between human judgments and MD-Score are reported in Figure 3 (left).

H Comparison with Memory Masking

We provide the prompt used in the pairwise comparison experiment between our SteeM and direct memory masking in Sections 5.3 in Figure 10. We also provide the user-simulator prompt used for memory-masking selection in Figure 10. The user simulator is powered by Gemini-2.5-Pro (Comanici et al., 2025).

Model	Method	RESEARCH				TUTORING				Avg. ↓
		Plan & Design	*Revise*	*Analyze & Critique*	*Concept Explanation*	*Plan & Design*	*Revise*	*Analyze & Critique*	*Concept Explanation*	
Qwen3-4B	Tag-cued (SFT)	1.10	1.50	1.11	0.90	1.29	1.47	1.38	0.88	1.20
	Tag-cued (SFT+RL)	1.01	1.48	1.08	0.84	1.28	1.43	1.35	0.82	1.16
	`SteeM` (SFT)	1.14	1.54	1.12	0.95	1.32	1.51	1.41	0.91	1.24
	`SteeM` (SFT+RL)	1.01	1.53	1.11	0.87	1.32	1.46	1.38	0.86	1.19
Qwen3-8B	Tag-cued (SFT)	0.99	1.36	1.06	0.85	1.26	1.49	1.28	0.84	1.14
	Tag-cued (SFT+RL)	0.97	1.34	1.05	0.82	1.27	1.45	1.27	0.82	1.12
	`SteeM` (SFT)	1.02	1.35	1.07	0.88	1.25	1.48	1.26	0.87	1.15
	`SteeM` (SFT+RL)	0.99	1.33	1.09	0.83	1.28	1.43	1.25	0.85	1.13

Table 6: Comparison on δ_{align} between training with tag-cued queries and NL-cued queries (`SteeM`). Lower is better.

Memory Dependence Rubric

1. Score Scale (1–5)

The rubric uses a uniform 1–5 scale across all dimensions to indicate how strongly an answer depends on project-/course-specific history, cross-session execution traces, and summarized preferences.

Overall meanings:

- **1 = Externalized / Generic Reconstruction.** The answer is reconstructed from generic domain principles; internal history serves only as loose topic cues.
- **2 = Lightly Contextualized / Ornamental Dependence.** History is referenced superficially and does not substantively drive content or reasoning.
- **3 = History-Aware / Integrated Dependence.** History meaningfully shapes content selection and prioritization; generic knowledge is filtered through the specific trajectory.
- **4 = History-Driven / Structural Dependence.** Internal artifacts define the backbone; past results/plans structurally constrain what is said.
- **5 = Continuation Mode / Deep Entrenchment.** The answer is a direct continuation of internal logs; understanding it requires access to specific history.

Usage note

- Scores must reflect how legally/structurally contingent the answer is on project-/course-specific history and internal artifacts.
- Judgments must be grounded in observable textual behaviors (content selection, reasoning structure, discourse style).
- Do *not* speculate about internal mechanisms.

2. Single Latent Axis: Project Memory Dependence

Name: Project Memory Dependence.

Short definition: degree to which the answer adheres to and extends the project/learner trajectory, rather than reconstructing a solution from generic principles.

Constraints:

- **Unidimensionality.** Content/Pattern/Style are projections of one latent axis; stronger orientation implies deeper reliance on internal artifacts and precedents.
- **Exclusion of aesthetic bias.** Do not incorporate independent style preferences (politeness, verbosity, optimism, etc.) except when they change insider vs. outsider stance.
- **Behavioral observability.** Base judgments only on the visible answer, query, and provided memory description (do not speculate about RAG/implementation).

3. Global Instructions

Goal: evaluate dependence along (1) Content selection, (2) Pattern & reasoning, (3) Stylistic stance. Dependence includes reuse/imitation/extension of internal materials: facts, execution summaries, error profiles, documented preferences.

Available: query, structured memory description, generated answer.

Ignore: general task quality unless incoherence prevents judging; ignore explicit meta-commentary; ignore length/politeness unless it changes insider vs. outsider stance.

N/A handling:

- If a diagnostic cue is unobservable, treat it as N/A; do not penalize missing artifacts that were never provided.
- Implicitly average over observable cues; final output is a single integer (1–5).

Scoring protocol:

Step 1: Context internalization (trajectory and available artifacts).
Step 2: Evidence marking (observable usage/non-usage cues).
Step 3: Dimension scoring (Content/Pattern/Style).
Step 4: Aggregation into `overall_memory_dependence_score`; Content/Pattern slightly higher than Style.
Step 5: Rationale (5–10 sentences citing specific textual evidence).

4. Dimensions

4.1 Content Axis — Content-Level Dependence

Definition: whether the substance (facts/examples/constraints/recommendations) is grounded in internal project materials rather than generic domain knowledge; whether core claims rely on specific artifacts (plans, results, feedback summaries) for validity.

Diagnostics:

- **Counterfactual test:** remove project memory $\Rightarrow$ do core claims remain justified?
- **Evidence basis:** are internal facts used as premises?
- **Artifact reuse:** substantive reuse of internal phases/directions/summaries?

Subdimensions: anchoring target; specificity/substitutability; artifact & summary reuse.

Levels:

- **Level 1 — Externalized.** Generic reconstruction; highly substitutable across similar projects.
- **Level 2 — Lightly contextualized.** Internal details are illustrative/minor constraints; core remains standard; artifacts loosely summarized.
- **Level 3 — History-aware.** History shapes scope/priorities; removing history makes key recommendations vague/unjustified.
- **Level 4 — History-driven.** Backbone defined by internal items; recommendations derived from past outcomes; heavy artifact reuse as building blocks.
- **Level 5 — Continuation mode.** Seamless continuation of internal logs; meaning opaque without specific memory; generic knowledge mostly connective.

4.2 Pattern Axis — Pattern-Level Dependence

Definition: whether organization/decomposition/reasoning aligns with established internal routes and documented preferences vs. generic external templates.

Diagnostics:

- **Process isomorphism:** replicate known internal workflow vs. impose standard template?
- **Reasoning continuity:** inherit criteria/trade-offs from past sessions?
- **Branching logic:** alternatives framed as controlled deviations vs. generic options?

Subdimensions: structural isomorphism; reasoning strategy continuity; alternative-path handling; cross-session process reuse.

Levels:

- **Level 1 — Generic pattern.** Standard framework; domain-general criteria; options in a vacuum.
- **Level 2 — Loosely echoing.** Occasional echoes; overall organization generic; cross-session mentions do not structure response.
- **Level 3 — Aligned pattern.** Internal routes integrated within accessible structure; options framed relative to the path.
- **Level 4 — Route-following.** Internal templates dominate; execution summaries serve as primary skeleton.
- **Level 5 — Process continuation.** Next step in idiosyncratic internal loop; unintelligible without route; options are micro-adjustments.

4.3 Style Axis — Style-Level Dependence

Definition: insider vs. outsider stance; continuity in shorthand/terminology/template language.

Subdimensions: context say/assume; terminology continuity; template-language reuse.

Levels:

- **Level 1 — External voice.** Standalone tutorial/report; neutral terminology; no insider shorthand/template reuse.
- **Level 2 — Lightly internalized.** Mostly external; occasional internal terms (often glossed); minimal template reuse.
- **Level 3 — Mixed voice.** Some shared background assumed; recognizable internal labels with partial reminders.
- **Level 4 — Insider collaboration.** Written for internal coordination; extensive unexplained shorthand; extensive template reuse.
- **Level 5 — Log-continuation.** Dense implicit context; discourse organized around internal naming schemes.

5. Joint Constraints

- All scores must be grounded in adherence to internal history/patterns/preferences; avoid unrelated factors.
- Treat unobservable cues as N/A; base scores only on evidence available; do not penalize absent artifacts not provided.
- **Weighting heuristic:** `overall_memory_dependence_score` driven primarily by Content + Pattern; Style is a modifier and should not shift the overall score by more than one level.

Table 7: Memory dependence judging rubric.

User-Simulator Prompt for Memory Masking

```
You are simulating a user in a long-horizon interaction setting.

You have a clear preference for how much the assistant should rely on historical memory when answering your query.
Your goal is to FILTER the available memory before the assistant responds, by DROPPING memory items that would push
the assistant away from your desired output.

Given:
(1) your query,
(2) your target memory-dependence score (your preference),
(3) the available memory items,

decide which memory items to drop so that the assistant's final response is most likely to match your preferred dependence
level.
Target dependence score: {score}
Score label: {score_label}

Rubric cues for score {score}:
- content_axis: {dim_desc.get('content_axis')}
- pattern_axis: {dim_desc.get('pattern_axis')}
- style_axis: {dim_desc.get('style_axis')}

Query:
{query}

Available memory items (each is optional; drop only when it is clearly unnecessary for the target score):
- user_profile: {mem['user_profile']}
- essential_artifacts_keys: {mem['essential_artifacts_keys']}
- past_5_events: {mem['past_5_events']}
- cross_session: {mem['cross_session']}

Decision guidelines:
- Optimize for the target dependence score (not for maximum helpfulness).
- Higher target score => keep more historically grounding content/pattern/style.
- Lower target score => drop memory that would anchor the response to historical details, templates, terminology, or
stylistic conventions.
- Keep memory items that are clearly required for correctness or task completion.
- If unsure, KEEP the item.
- Be conservative: drop the minimum necessary to reach the target.

Return JSON ONLY, following exactly this schema (include all fields):
{
  "drop": {
    "user_profile": true|false,
    "essential_artifacts_keys": [list of keys to drop],
    "past_5_events_ids": [list of event_id to drop],
    "cross_session": true|false
  },
  "reason": "one-sentence rationale focusing on why these drops help match the target score"
}
```

Figure 9: User-Simulator prompts for memory masking.

Prompt for Pairwise Comparison

You are a pairwise judge. Given the user query, memory context, and two model answers (A and B), pick the better answer.

```
########################################
######## MEMORY (full_context) ########
########################################
{memory}

##########################
######## ANSWER A ########
##########################
{answer_a}

##########################
######## ANSWER B ########
##########################
{answer_b}

############################
######## USER QUERY ########
############################
{query}

######## TARGET DEPENDENCE PREFERENCE ########
score_label: {label}
- content_axis: {pref['desc'].get('content_axis')}
- pattern_axis: {pref['desc'].get('pattern_axis')}
- style_axis: {pref['desc'].get('style_axis')}
```

Now carefully judge the two answers based on how well they align with the target dependence preference. Choose the answer that better meets the target dependence level indicated.
Return strict JSON: {{"winner": "A" | "B", "reason": "short rationale"}}. Choose one of A/B.

Figure 10: Prompt for pairwise comparison.

Protocol for Human Pairwise Annotation of Memory Reliance

You are a human annotator. You will be shown:
(1) the user query,
(2) the provided memory/history context,
(3) two candidate answers (A and B).

Your task:
Choose which answer relies MORE on the provided memory/history to produce its response.

```
################################
########## MEMORY #############
################################
{memory}

################################
######### ANSWER A ############
################################
{answer_a}

################################
######### ANSWER B ############
################################
{answer_b}

################################
######## USER QUERY ###########
################################
{query}
```

How to judge "relies more on memory":
- An answer is MORE memory-reliant if its main points depend on specific details in MEMORY (e.g., prior decisions/results/constraints/preferences/established terminology), or if it reads like a continuation of an ongoing project/lesson that would be hard to write correctly without MEMORY.
- An answer is LESS memory-reliant if it is mostly a generic standalone response that would still be appropriate for many similar users/projects even if MEMORY were removed.
- Superficial name-dropping of MEMORY without affecting the substance counts as weak reliance.

What to ignore:
- General correctness/helpfulness, unless correctness hinges on a specific fact from MEMORY.
- Meta-statements like "I used your history…" if they do not actually change the content.

If both seem similar:
Pick the one that depends more on MEMORY-specific details (i.e., would lose justification or specificity without MEMORY).
No tie option.

Figure 11: Protocol for human pairwise annotation of memory reliance.