PRAMEM: PRACTICE-DERIVED EXPERIENTIAL MEMORY FOR LONG-HORIZON BEHAVIOR PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon behavior prediction plays a crucial role in artificial intelligence application, which aims to infer a user's next action based on a lengthy historical sequence. The rise of large language models (LLMs) offers a promising direction for sequential behavior prediction, but LLMs still struggle with latent behavioral pattern induction and model-intrinsic cognitive biases when tackling long-horizon behavior prediction. Prior memory management methods follow a context-compression paradigm that attempts to address this task by alleviating the historical sequence burden, yet fail to resolve core challenges. In this paper, we advocate a paradigm shift that reframes the lengthy historical sequence from a burden into a valuable resource to be exploited, and accordingly propose PraMem, which conducts practice over lengthy historical sequence to build an experiential memory, thereby serving as assisted input for accurate long-horizon behavior prediction. Extensive experiments across diverse tasks demonstrate PraMem achieves superior performance than prior methods, and more in-depth analyses provide valuable insights into mechanism of the experiential memory.

1 INTRODUCTION

Long-horizon behavior prediction aims to unbiasedly infer a user's action in current scene by inducing latent behavioral patterns from a lengthy historical sequence of scene-action records, serving as a critical foundation for recommendation systems, cognitive science, and interactive artificial intelligence (McClelland, 2009; Petrović, 2018; Ren et al., 2019; Pi et al., 2019). Recently, formulating sequential behavior prediction as an autoregressive generation task with large language models (LLMs) has emerged as a promising direction, but long-horizon sequences spanning thousands of steps still pose a substantial challenge (Zhang et al., 2024; Chen et al., 2026). On one hand, behavioral pattern induction from ultra-long sequences is highly difficult, as fine-grained factors influencing the next action (e.g. *users' preferences for brand, price, quality, and appearance when selecting products*) are hidden and scattered across lengthy sequence, making them hard to capture accurately (Lin et al., 2024; Liu et al., 2025). On the other hand, LLMs are subject to model-intrinsic cognitive biases when performing this task, such as *tendency to follow majority, prejudice from training, and disproportionate attention to certain records in sequence*, all of which further degrade performance on long-horizon behavior prediction (Liu et al., 2024; Weng et al., 2025).

To enhance LLMs on long-horizon behavior prediction tasks, previous studies have explored memory management to alleviate the long-context burden through information compression, which removes redundant text of the historical sequence via extraction and aggregation, and leverages retrieval to retain only the most relevant behavioral records for prediction (Zheng et al., 2024; Wang et al., 2024; Xi et al., 2024; Wang et al., 2025a). However, such a context-compression paradigm fails to effectively address the core challenges of long-horizon behavior prediction. First, information compression is not equivalent to the behavioral pattern induction. LLMs still need to induce behavioral patterns on the fly from fragmented behavioral records during the prediction stage, which makes the reliability of induced behavioral patterns hard to guarantee (Li et al., 2025a; Jiang et al., 2025). Second, previous methods focus solely on memory management over the historical sequence, lacking any mechanism to perceive or correct model-intrinsic cognitive biases of the LLM. This means that even when the provided information is sufficiently accurate and relevant, intrinsic biases of the LLM can still lead to incorrect prediction (Weng et al., 2025; Sumita et al., 2025).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

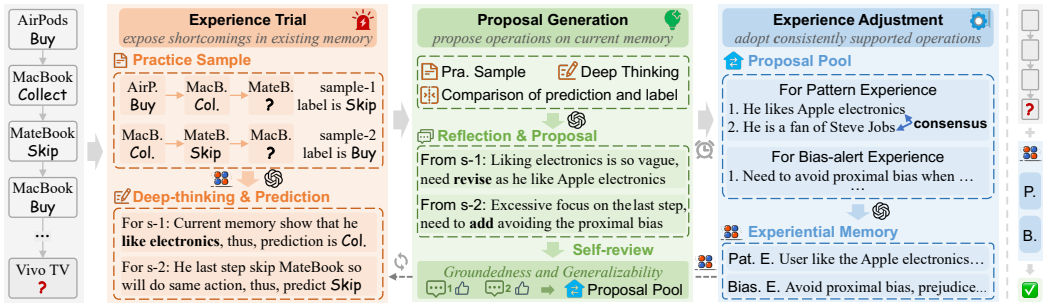

Figure 1: PraMem dynamically maintains the pattern experience and bias-alert experience by iteratively performing existing experience trial, reflective proposal generation, and consensual experience adjustment, and finally uses experiential memory for accurate long-horizon behavior prediction.

Limitations of above context-compression paradigm stem from the guiding philosophy, which treats historical sequences mainly as long-context burden to alleviate. In contrast, dialectical philosophy suggests that burden and resource are a unity of opposites, where the two can transform into each other under the right condition (Engels, 1960; Zedong, 1987). In this task, while lengthy historical sequences impose a heavy burden, they also contain valuable resources for improving LLM performance, as each segment of historical sequence naturally forms a practice sample for sequential behavior prediction with a ground-truth label. Therefore, dialectical philosophy can inspire a new paradigm, instead of passively alleviating the burden of historical sequences through compression, LLMs can proactively leverage these valuable resources for practice. Through iterative trial-and-error, LLMs can induce behavioral patterns in historical sequence and guarantee their reliability, as well as find out model-intrinsic biases for the user's behavior prediction and alert against them.

Inspired by above ideas, we propose **PraMem**, a training-free framework that builds a time-evolving experiential memory by beforehand practice on historical sequence, thereby assisting accurate prediction. Specifically, as illustrated in Figure 1, the experiential memory consists of pattern experience for presenting user behavioral patterns and bias-alert experience for alerting LLM intrinsic biases. In order to dynamically maintain the experiential memory, PraMem iteratively performs existing experience trial, reflective proposal generation, and consensual experience adjustment. First, to expose shortcomings in existing experience, PraMem draws segments from historical sequence to construct practice samples with ground-truth labels, and prompts the LLM to predict via deep thinking under current experiential memory. Second, to revise, prune, or supplement current experiential memory, PraMem performs reflection on the practice sample, thinking process, and prediction-label comparison, generating a batch of reflective proposals, from which reliable ones are retained via a self-review mechanism and added to a proposal pool. Finally, to prevent unstable adjustment of experiential memory caused by occasional behaviors, PraMem performs consensus-driven adjustment based on the proposal pool at fixed round intervals, where only operations consensually supported by multiple proposals are adopted, enabling stable evolution of experiential memory.

Within the above framework, the quality of reflective proposals is a key factor in ensuring the overall effectiveness, requiring each proposal to be both grounded in and generalizable beyond the practice sample. To this end, we design a self-review mechanism to filter out unreliable proposals by applying transformations to the practice sample. First, to identify proposals without sufficient groundedness with respect to the practice sample, we perturb the historical sequence of the sample so that its underlying behavioral patterns change significantly. If the original proposal still holds after the perturbation, it indicates that the proposal is not genuinely grounded in the practice sample. Second, to identify proposals that lack generalizability, we rewrite the prediction scene of the practice sample to generate multiple similar but distinct virtual scenes, forming a multiple-choice task of scenes. If a proposal can unambiguously guide the correct selection of the true scene, it indicates that the proposal is overly specific to the practice sample rather than generally applicable. Through this self-review mechanism, only reflective proposals that are both well-grounded and generalizable are added to the proposal pool, thereby ensuring the overall effectiveness of PraMem.

In experiments, we evaluate PraMem across various long-horizon behavior prediction tasks and compare it with several strong memory management baselines, the results demonstrate that PraMem achieves superior performance, confirming PraMem is an effective solution for long-horizon behav-

ior prediction. Additionally, comprehensive ablation studies verify the effectiveness of each core component as well as the self-review mechanism, while more detailed analysis further illustrate the evolution of experiential memory throughout practice process and reveal the important roles of the elaborate designs in the PraMem. The main contributions of this paper can be summarized as:

- We propose PraMem, which conducts iterative practice over the lengthy historical sequence to build an experiential memory, thereby assisting for accurate long-horizon behavior prediction.
- We design the self-review mechanism, which can filter out reflective proposals without sufficient groundedness and good generalizability, thereby ensuring the overall effectiveness of PraMem.
- We conduct extensive experiments and analysis, which confirm the advantage of PraMem and provide several valuable insights into the evolution and mechanism of the experiential memory.

2 PRAMEM

2.1 TASK FORMULATION

Vanilla Process. From the perspective of a single user, the LLM is employed to predict the user's action $\hat{a}_t$ in a current scene s_t based on the user's lengthy historical sequence of scene-action records:

$$\hat{a}_t = \mathrm{LLM}(\mathcal{H}_t,\, s_t)\,, \quad \mathcal{H}_t = \{(s_i, a_i)\}_{i=1}^{t-1} \tag{1}$$

where $\mathcal{H}_t$ denotes the historical sequence up to step t, with s_i and a_i representing the scene and action at step i. All scenes and actions are represented in natural language, and the LLM generates $\hat{a}_t$ in an autoregressive manner conditioned on the concatenation of $\mathcal{H}_t$ and s_t.

PraMem Process. To boost LLMs on long-horizon behavior prediction, PraMem performs beforehand practice over the lengthy historical sequence to build a user-specific and time-evolving experiential memory $\mathcal{M}_t$, which can assist the LLM in accurate prediction, as following formula:

$$\mathcal{M}_t = \mathrm{Practice}(\mathcal{H}_t,\, \mathcal{M}_{t'})\,, \quad \hat{a}_t = \mathrm{LLM}\Big(\tilde{\mathcal{H}}_t,\, s_t,\, \mathcal{M}_t\Big)\,, \quad \tilde{\mathcal{H}}_t = \{(s_i, a_i)\}_{i=t-k}^{t-1} \tag{2}$$

where $\mathcal{M}_{t'}$ $(t' < t)$ denotes the previous state of the experiential memory, based on which $\mathcal{M}_t$ is incrementally evolved by practicing on the newly accumulated records, and $\tilde{\mathcal{H}}_t$ is a short suffix containing only the k most recent records adjacent to the current prediction step ($k \ll t$).

2.2 PRACTICE-DERIVED EXPERIENTIAL MEMORY

To address the two core challenges of LLM-based long-horizon behavior prediction, PraMem dynamically maintains an experiential memory that consists of two types of experience:

- **Pattern Experience (Pat. E.)**: Presenting user behavioral patterns embedded in lengthy historical sequence, for example, *this user likes Apple's electronic products, this user buys daily necessities at the beginning of each month, this user tends to add items to favorites before purchase*.
- **Bias-alert Experience (Bias. E.)**: Alerting the LLM to intrinsic biases that are likely to arise when handling the current user, for example, *avoiding the proximal bias that only considers recent behavioral records, avoiding the prejudice assumption that Asians prefer some mysticism things*.

With this twofold experiential memory, the LLM can acquire reliable behavioral patterns of the user and stay alert to its own error-prone biases before making the formal prediction, thereby enabling more accurate long-horizon behavior prediction. Notably, the experiential memory is user-specific and time-evolving: as the user's behavioral sequence continues to grow, PraMem dynamically maintains the experiential memory by iteratively performing existing experience trial, reflective proposal generation, and consensual experience adjustment, so that the memory becomes increasingly precise in capturing behavioral patterns and increasingly pertinent in signaling cognitive biases.

Existing Experience Trial. In order to sufficiently expose shortcomings of the existing experience, PraMem samples q segments from the historical sequence to construct behavior prediction practice samples with ground-truth labels, and then prompts the LLM to carefully do explicit deep thinking under the current experiential memory so as to fully leverage the existing experience for prediction. Specifically, let $\mathcal{M}_{t'}$ denote the current state of the experiential memory, which has been built upon

the historical sequence $\mathcal{H}_{t'}$ up to step t'. Taking a single sample as an illustration:

$$(\tilde{s}_j, \tilde{a}_j) \sim \mathcal{H}_t \setminus \mathcal{H}_{t'}, \quad \tilde{\mathcal{H}}_j = \{(s_i, a_i)\}_{i=j-k}^{j-1}, \quad (\hat{a}_j, r_j) = \mathrm{LLM}_{\text{thinking}}\Big(\tilde{\mathcal{H}}_j,\, \tilde{s}_j,\, \mathcal{M}_{t'}\Big) \tag{3}$$

where $(\tilde{s}_j, \tilde{a}_j)$ denotes the sampled target scene and ground-truth label, with j later than the time point t' covered by the current experiential memory (i.e., $t' < j < t$), ensuring that the evolution of the experiential memory remains consistent with the temporal order of the user's behavioral sequence, $\tilde{\mathcal{H}}_j$ is the short historical suffix preceding the target scene, which together with $\tilde{s}_j$ constitutes the complete practice sample, k is a hyperparameter controlling the length of $\tilde{\mathcal{H}}_j$, $\hat{a}_j$ and r_j are respectively the predicted action and the explicit deep-thinking process generated by the LLM under the guidance of the current experiential memory. Prompts in this part are shown in Appendix C.

Reflective Proposal Generation. After the above trial on the sampled practice sample, PraMem performs reflection by jointly considering the practice sample, deep-thinking process, and comparison between predicted action and ground-truth label, thereby producing proposals for revising, supplementing, or pruning the experiential memory. Specifically, each proposal is composed of the type of experience it targets and detailed content describing the specific operation. To ensure the reliability of the produced proposals, a self-review mechanism is further introduced to filter out the unreliable ones, and only those passing the review are accepted into a proposal pool that supports the subsequent adjustment of the experiential memory. The process can be expressed as following:

$$\{p_i\}_{i=1}^{n} = \mathrm{LLM}_{\text{reflection}}\Big(\tilde{\mathcal{H}}_j,\, \tilde{s}_j,\, \tilde{a}_j,\, \hat{a}_j,\, r_j\Big), \quad \mathcal{P} \leftarrow \mathcal{P} \cup \{p_i \mid \mathrm{SelfReview}(p_i) = \text{Pass}\}_{i=1}^{n} \tag{4}$$

where p_i denotes the i-th proposal generated by the LLM after careful reflection, and n is a hyperparameter specifying the number of generated proposals. $\mathrm{SelfReview}(\cdot)$ judges whether a proposal is well-grounded and generalizable, and $\mathcal{P}$ is the proposal pool that accumulates all reliable proposals across rounds for the subsequent experience adjustment. The detailed design of self-review mechanism is presented in Section 3 and detailed prompts used in this part are shown in Appendix D.

Consensual Experience Adjustment. In the iterative loop, the above existing experience trial and reflective proposal generation are executed cyclically to accumulate the proposal pool, while the experience adjustment is triggered every T rounds to avoid instability caused by occasional user behaviors in the historical sequence. Specifically, when triggered, PraMem utilize the LLM to perform consensus-driven adjustment of experiential memory based on the proposal pool, where only operations consistently supported by multiple proposals are adopted, enabling stable evolution of experiential memory. The consensus-driven adjustment process can be formulated as following:

$$\mathcal{M}_t,\, \mathcal{P} = \mathrm{LLM}_{\text{consensus}}(\mathcal{M}_{t'},\, \mathcal{P}) \tag{5}$$

where $\mathcal{M}_t$ is the updated experiential memory produced by the LLM through identifying and applying the consistently supported operations in $\mathcal{P}$. After the adjustment, the proposals that have been adopted are removed from $\mathcal{P}$, while the remaining ones are retained for accumulation in the subsequent iterative loop. Detailed prompts used in this part are shown in Appendix E.

3 Self-review Mechanism

In the above PraMem, reflective proposals directly dictate the generation and evolution of experiential memory, the quality of reflective proposals serves as a critical determinant of overall effectiveness. On one hand, proposals must be well-grounded in the practice sample rather than being fabricated by the LLM, thereby ensuring the resulting experiential memory faithfully captures authentic requirements for prediction. On the other hand, proposals must be sufficiently generalizable rather than being overly specific to a single practice sample, thereby enabling the experiential memory to effectively guide predictions on unseen scene. To this end, we introduce a self-review mechanism to sequentially conduct groundedness review and generalizability review, which can effectively filter out proposals that lack substantial groundedness or possess poor generalizability, as in Figure 2.

Groundedness Review. To identify proposals that lack groundedness with respect to the practice sample, we leverage the LLM to perturb the historical sequence of the sample such that its underlying behavioral patterns are substantially altered. Then the LLM is employed to assess whether the

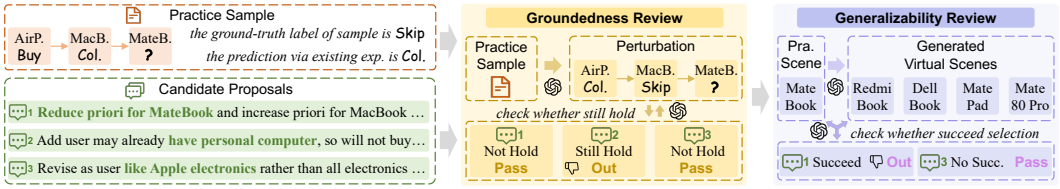

Figure 2: Self-review mechanism only the retains reliable proposals that are both grounded in and generalizable beyond the practice sample, thereby ensuring the overall effectiveness of PraMem.

original proposal still holds on the perturbed practice sample. If still holding, it indicates the proposal is not genuinely grounded in the practice sample. This process can be formulated as follows:

$$\tilde{\mathcal{H}}'_j = \text{LLM}_{\text{perturb}}(\tilde{\mathcal{H}}_j), \quad \text{Gr.Review}(p_i) = \begin{cases} \text{Out}, & \text{if LLM}_{\text{judge}}(p_i \mid \tilde{\mathcal{H}}'_j,\, \tilde{s}_j,\, \tilde{a}_j) = \text{Hold} \\ \text{Pass}, & \text{otherwise} \end{cases} \tag{6}$$

where $\tilde{\mathcal{H}}'_j$ is the perturbed historical sequence, in which the actions are deliberately rewritten by the LLM to substantially change the behavioral pattern. Prompts for this part are in Appendix F.

Generalizability Review. To identify proposals that lack generalizability, we leverage the LLM to rewrite the prediction scene of sample, generating virtual scenes that share the same category as the true scene but differs in fine-grained attributes, thereby constituting a multiple-choice task. Then the LLM is employed to determine whether the proposal can unambiguously guide selection of true scene. If so, the proposal is overly specific to the current practice sample. The formula is as follows:

$$\{s_v^{(i)}\}_{i=1}^{m} = \text{LLM}_{\text{rewrite}}(\tilde{s}_j), \quad \text{Ge.Review}(p_i) = \begin{cases} \text{Out}, & \text{if LLM}_{\text{select}}(\mathcal{S}_j \mid p_i) = \tilde{s}_j \\ \text{Pass}, & \text{otherwise} \end{cases} \tag{7}$$

where $\{s_v^{(i)}\}_{i=1}^{m}$ denotes the m virtual scenes based on the true scene $\tilde{s}_j$, and $\mathcal{S}_j = \{\tilde{s}_j\} \cup \{s_v^{(i)}\}_{i=1}^{m}$ is the scene set forming the multiple-choice task. Prompts for this part are in Appendix G.

4 Experiments

4.1 Experimental Settings

Datasets. We select three publicly available and widely recognized benchmarks that encompass diverse long-horizon behavior prediction scenarios, enabling a comprehensive and rigorous evaluation. (1) **OmniBehavior** (Chen et al., 2026) contains long historical sequences of users from the Kuaishou platform, requiring behavior prediction across four scenarios: video browsing, live streaming, advertisement, and e-commerce. (2) **MemoryCD** (Zhang et al., 2026a) is constructed from the real-world Amazon Reviews 2023 corpus. We adopt its cross-domain setting, where each user is characterized by years of review interactions spanning four domains: books, electronics, home & kitchen and personal care. (3) **MovieLens-1M** (Harper & Konstan, 2015) is collected from MovieLens, a movie recommendation system maintained by the University of Minnesota since 1997. It records users' chronological ratings of movies on a 5-point scale. All three benchmarks share the same evaluation metric, accuracy (ACC). Specifically, OmniBehavior is a binary classification task, whereas MemoryCD and MovieLens-1M are 5-class rating-prediction tasks.

Baselines. We select some commonly used methods as well as advanced memory management approaches for long-horizon behavior prediction tasks to conduct comprehensive comparisons with PraMem. (1) **No Memory Management**. This category includes methods that directly set the LLM's input window to 32K tokens (Long-context) to accommodate as much of the long historical sequence as possible, as well as Truncation methods that retain only the most recent records from the long historical sequence. (2) **Simple Memory Management**. This category includes the RAG method (Lewis et al., 2020) that uses a retriever to recall the most relevant records as memory from the long historical sequence with respect to the prediction scene, and summary method (Wang et al., 2023b) that constructs memory through iterative summarization based on the long historical sequence. (3) **Advanced Memory Management**. This category includes Mem0 (Chhikara et al., 2025) and MemOS (Li et al., 2025b), which leverage LLMs to perform information extraction and summarization on the long historical sequence and use a retriever to recall the most relevant text as

Category	Method	OmniBehavior				MemoryCD				ML-1M
		Video	Live	Ads	E-comm.	Books	Elec.	Home	Person.	Movie
Backbone: GPT-OSS-120B										
No Memory	Long-context	74.9	67.2	59.4	72.0	60.8	61.8	62.5	60.0	40.6
	Truncation	79.5	73.2	61.0	80.1	61.1	62.8	64.1	58.7	38.7
Simple Memory	RAG	77.9	71.6	61.3	65.7	51.7	52.2	57.2	55.9	34.9
	Summary	80.1	77.0	64.7	74.5	57.1	63.7	63.8	59.1	40.4
Advanced Memory	Mem0	81.7	75.0	63.1	78.9	62.9	61.4	62.5	60.1	38.7
	MemOS	78.9	74.4	61.8	77.3	60.3	61.0	61.6	59.9	38.8
	ProEx	82.5	77.1	68.1	83.3	61.6	62.3	60.7	**60.2**	37.7
Experiential Memory	**PraMem (Ours)**	**86.3***	**84.4***	**80.8***	**85.3***	**63.8***	**64.7***	**64.4***	59.7	**42.9***
Backbone: Qwen3.5-35B-A3B										
No Memory	Long-context	61.5	62.0	55.7	62.2	59.0	61.2	65.7	57.8	37.8
	Truncation	67.7	64.3	58.0	65.3	59.8	60.7	66.2	60.5	37.1
Simple Memory	RAG	64.2	60.5	55.9	61.3	55.9	58.2	62.5	58.2	30.6
	Summary	67.3	65.6	59.4	64.2	54.6	54.4	61.3	55.0	36.5
Advanced Memory	Mem0	67.2	66.6	55.6	65.9	60.2	60.8	65.9	61.5	36.3
	MemOS	65.8	65.6	54.2	67.9	59.7	60.5	66.3	58.8	37.7
	ProEx	70.7	69.0	61.3	69.2	60.7	61.5	66.0	60.0	38.0
Experiential Memory	**PraMem (Ours)**	**77.8***	**74.8***	**75.4***	**75.4***	**61.4***	**62.2***	**67.4***	**61.9***	**39.8***

Table 1: Main results across two backbones and three public benchmarks. PraMem is effective for long-horizon behavior prediction, and the experiential memory can transfers well across backbones. In above table, * indicates statistically significant improvements ($p < 0.01$) over baselines.

memory, as well as ProEx (Zhang et al., 2026b), which employs LLMs to induce user behavioral patterns from the long historical sequence to generate multiple dynamic profiles for each user, and refines them by cross-validation across profiles to obtain the most accurate user profile as memory.

Implementation Details. The experiential memory is stored locally, with each user corresponding to a JSON file containing three lists for pattern experience, bias-alert experience, and the proposal pool, where each experience or proposal is stored as a list element. At initialization, the pattern experience is filled with basic user information explicitly provided in the benchmark's historical sequence, the bias-alert experience with generic prompts on common LLM biases, and the proposal pool is left empty. For memory maintenance, we use GPT-OSS-120B[1] as the LLM for all PraMem procedures. We set n in Eq. 4 to 10, m in Eq. 7 to 7, the adjustment interval T to 5, and the per-round practice sample number q to 1, and run 80 rounds for each user. All practice samples chronologically precede the prediction target, ensuring that neither its label nor any future information is used. Memory updates are performed through structured add/modify/delete operations on ID-indexed entries rather than by regenerating the entire memory. For evaluation, Long-context uses a 32K-token input window, while Truncation retains the most recent k records within 8K tokens, with k defined as in Eq. 2. All other baselines and PraMem likewise use only these k records. For fairness, the memory is uniformly concatenated with the truncated historical sequence as the joint input for PraMem and all memory-based baselines. We evaluate two backbones, GPT-OSS-120B[1] and Qwen3.5-35B-A3B[2], both deployed via vLLM[3] with default parameters for supporting reproducibility.

4.2 Main Results

Results compared with baselines are shown in Table 1, there are two main conclusions:

PraMem is a powerful method for addressing long-horizon behavior prediction tasks. For example, PraMem improves the ACC from 79.5 (Truncation baseline) to 86.3 on the Video subset of OmniBehavior. In contrast, prior advanced memory management methods yield marginal gains. Mem0, which relies on LLM-based complex extraction, aggregation, and retrieval, achieves only 81.7. And ProEx, which performs memory management via dynamic user profiles generated by LLMs, achieves only 82.5. Overall, across a total of 9 long-horizon behavior prediction scenar-

[1] https://huggingface.co/openai/gpt-oss-120b
[2] https://huggingface.co/Qwen/Qwen3.5-35B-A3B
[3] https://pypi.org/project/vllm/

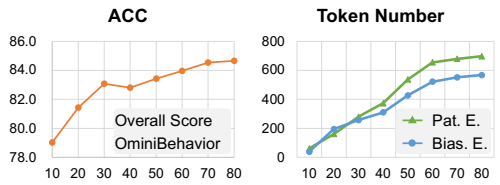

Figure 3: Memory evolution. The overall performance improves with more practice rounds, while the average token number of experiential memory first increases and finally stabilizes.

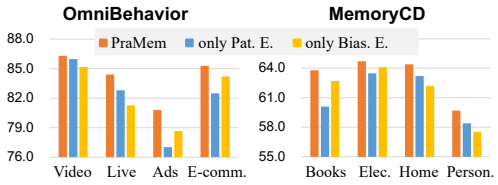

Figure 4: Effectiveness of twofold experience. It shows that both of the pattern experience and the bias-alert experience play positive roles for assisting LLM long-horizon prediction.

ios spanning OmniBehavior, MemoryCD, and ML-1M benchmark, PraMem outperforms all the baseline methods on the vast majority of scenarios, collectively demonstrating that practice-derived experiential memory constitutes an effective solution for this class of challenging problems.

The practice-derived experiential memory generalizes across different LLMs, demonstrating good practicality. As described in the implementation details, we employ only GPT-OSS-120B for constructing experiential memory, while adopting two backbones during evaluation. As shown in Table 1, the experiential memory is effective not only on GPT-OSS-120B but also yields performance gains when the backbone is switched to Qwen3.5-35B-A3B. This observation aligns with design principles and prior findings. The pattern experience primarily presents user behavioral patterns, which are naturally transferable across backbones. As for the bias-alert experience, prior studies have revealed that different LLMs share certain common internal cognition (Huh et al., 2024), so that bias-alert experience derived from one LLM remains effective for another. This generalizability offers good practical value, eliminating the need for model-specific reconstruction.

4.3 Evolution of Experiential Memory

To gain deeper insights into the role of experiential memory and its evolution throughout the iterative practice process, we present in Figure 3 the performance achieved using experiential memory from different practice rounds, along with statistics on the average token number of experiential memory. There are two conclusions about evolution of experiential memory during practice process:

The performance of experiential memory on long-horizon behavior prediction progressively improves with more practice rounds. As shown in the leftmost subplot of Figure 3, using the experiential memory obtained after 10 practice rounds can achieve an overall score of 79.0, which further improves to 83.5 at the 40th round, and reaches 84.7 at the 80th round. This monotonically increasing overall score with respect to the number of practice rounds suggests that the experiential memory undergoes a progressive evolution throughout the iterative practice process, continuously enhancing its utility for long-horizon behavior prediction. This phenomenon is consistent with our design, where iterative trial-and-error enables the gradual derivation of a good experiential memory.

During evolution, the total length of experiential memory first increases and then stabilizes. As shown in the rightmost subplot of Figure 3, the total length of experiential memory gradually increases from round 10 to round 60, and then stabilizes after round 60, while the overall performance exhibits a steady upward trend throughout the entire process. This is because PraMem adjusts the experiential memory through three types of operations (i.e. revising, supplementing, and pruning). After a certain number of rounds, the memory achieves relatively comprehensive coverage, and subsequent iterations focus on refining and correcting the existing experience rather than continuously adding new information. This further demonstrates the practical value of PraMem: unlike conventional memory management methods that always keep accumulating information, PraMem avoids length explosion as rounds increase, thereby preventing additional long-context burdens on LLM.

4.4 Detailed Analysis

Effectiveness of Twofold Experience. To validate in detail the contributions of the two types of experience in PraMem (i.e., pattern experience and bias-alert experience), we report in Figure 4 the performance achieved when using only one type of experience at a time. The results show that

Configuration	OmniBehavior				MemoryCD				ML-1M
	Video	Live	Ads	E-comm.	Books	Elec.	Home	Person.	Movie
PraMem	**86.3**	**84.4**	**80.8**	**85.3**	**63.8**	**64.7**	**64.4**	**59.7**	**42.9**
w/o deep-thinking trial	80.2	79.9	76.5	81.8	61.7	64.1	60.2	56.5	38.7
w/o reflective proposal	84.7	82.0	77.6	80.9	62.1	63.1	63.8	58.0	36.9
w/o consensual adjustment	85.9	84.0	78.3	84.1	60.8	64.5	62.9	57.4	39.4

Table 2: Ablation results of the three core modules. Deep thinking in experience trial, reflection in proposal generation, and consensus setting in experience adjustment all play crucial positive effect.

Raw Proposal Quality		*Self-review Reliability*	
	Ratio (%)		Ratio (%)
Bad General.	23.5	Prec.	81.7
Bad Ground.	20.5	Recall	78.2

Table 3: Necessity and reliability of self-review. Raw proposals contain some bad cases and the self-review is a good filtering mechanism.

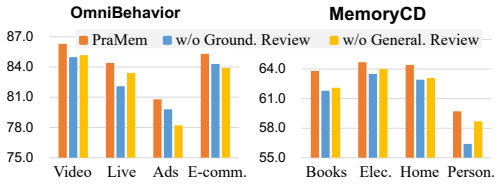

Figure 5: Effectiveness of the self-review mechanism. It shows that self-review mechanism can effectively ensure the final performance.

removing either type of experience leads to a clear performance decline, regardless of which one is removed. This finding, to some extent, confirms a notable limitation of prior memory management methods: by solely considering the historical sequence while neglecting the intrinsic biases of the LLM, such methods fail to fully capture the factors that jointly determine prediction quality. In contrast, *PraMem jointly considers pattern experience, which presents user behavioral patterns, and bias-alert experience, which alerts against LLM-intrinsic biases, thereby representing a more advanced and comprehensive strategy for long-horizon behavior prediction*.

Ablation Results. To investigate the necessity of several key designs in PraMem, we conduct ablation studies, with results reported in the above Table 2. Specifically, three variants are considered. *w/o deep-thinking trial"* removes deep thinking in experience trial and instead makes predictions directly, without this step, the model fails to adequately expose its shortcomings. *w/o reflective proposal"* removes reflection in proposal generation and instead generates proposals directly from the prediction-label comparison of the practice sample, this tends to produce inaccurate proposals. *"w/o consensual adjustment"* removes the consensus setting in experience adjustment and instead directly adopts all proposals, this degrades the quality of experiential memory due to the influence of occasional behaviors. As shown in the table, all three components, *deep thinking in experience trial, reflection in proposal generation, and consensus setting in experience adjustment, play crucial positive roles in the overall performance of PraMem for long-horizon behavior prediction.*

Analysis of Self-review Mechanism. To in-detail validate contributions of the self-review mechanism in PraMem (i.e., groundedness review and generalizability review), we investigate it from necessity, reliability, and effectiveness. As reported in Table 3, which summarizes a manual audit of sampled proposals, the raw proposals generated by the LLM are not always trustworthy, 23.5% of them fail in generalization and 20.5% lack groundedness, confirming the necessity of filtering out unreliable proposals. Self-review performs this filtering accurately, achieving a precision of 81.7% and a recall of 78.2%, indicating that it rejects most genuinely flawed proposals while rarely discarding valid ones. As shown in Figure 5, removing the self-review mechanism yields a clear performance drop, demonstrating that *the self-review mechanism effectively identifies and filters out unreliable proposals, thereby ensuring the quality of the experiential memory*.

Case Study. To intuitively demonstrate the advantages of the experiential memory built by PraMem over traditional memory management methods, we conduct a case study that compares the memory maintained for the same user under PraMem and the representative baseline MemOS, as shown in Appendix H. This case reveals a fundamental difference between the two: *the baseline's memory is a record of facts, remaining at the level of what has happened, in contrast, PraMem's memory is the explicit and concise experience distilled through iterative practice*, which can effectively presents the user's latent behavioral patterns and pre-alerts the LLM against its systematic biases, thereby genuinely resolving the core challenges of long-horizon behavior prediction for better performance.

Efficiency Report. We report in Table 4 the average per-user time cost of memory construction for PraMem and three advanced memory-based methods. Since PraMem performs extensive iterative practice over the historical sequence, it naturally incurs higher construction overhead than conventional compression-based methods, while the overall cost remains within an acceptable range. Moreover, in practical applications, memory construction can be conducted offline, and thus does not increase the latency of online prediction. In this sense, PraMem trades a moderately larger offline construction cost for substantial performance gains, which we believe is acceptable. We further discuss the efficiency issue in detail in the Limitations A.

	Construction Time (min)	Overall ACC
Mem0 (Chhikara et al., 2025)	34.1	60.0
MemOS (Li et al., 2025b)	30.6	60.7
ProEX (Zhang et al., 2026b)	24.7	61.8
PraMem (Ours)	28.8	66.2

Table 4: Efficiency report of PraMem and memory-based baselines. It shows that PraMem trades a moderately larger offline construction cost for good performance gains.

5 Related Work

Experiential Memory. Experiential memory, as a core mechanism for self-evolution in agents, has been widely adopted in the agent domain (Hu et al., 2025; Gao et al., 2025). The core idea is to encode historical interaction trajectories, successful experiences, and failure lessons into durable, retrievable knowledge representations, enabling agents to reuse such experiences in subsequent tasks to avoid repeating mistakes and to improve decision quality (Zheng et al., 2023; Zhang et al., 2025). Specifically, some methods consolidate experience into executable procedural capacities such as building ever-growing skill libraries (Wang et al., 2023a; Zheng et al., 2025; Bouzenia et al., 2025; Fang et al., 2025). Other methods distill transferable reasoning patterns and high-level insights from past trajectories to guide subsequent decisions (Ouyang et al., 2025; Li et al., 2026; Zhu et al., 2026; Yang et al., 2026). Long-horizon behavior prediction tasks inherently contain abundant practice samples, providing an ideal data foundation for constructing experiential memory. Therefore, introducing experiential memory mechanisms into this task holds great promise and potential.

Memory for Behavior Prediction. Positioning our work within memory management, we discuss related memory-based approaches. To assist LLMs in long-horizon behavior prediction, prior works have primarily focused on memory management through extraction, summarization, and retrieval over historical sequence, thereby alleviating long-context burden of LLMs (Zheng et al., 2024; Wang et al., 2024; Xi et al., 2024; Wang et al., 2025a; Huang et al., 2025), but do not directly address the core challenges of long-horizon behavior prediction. More recently, some works have attempted to dynamically construct user profiles as a form of memory, such as ProEx and LettinGo build comprehensive user profiles from multiple aspects by jointly considering the historical sequence through multi-granularity consistency checks (Wang et al., 2025b; Zhang et al., 2026b), and DGDPO performs diagnostic evaluation on the user profile by sampling segments from the historical sequence, so as to identify and correct deficiencies in the current profile (Liu et al., 2026). Nevertheless, these works focus solely on memory management over the historical sequence and overlook the intrinsic deficiencies exhibited by LLMs when performing the task, and they lack a review mechanism for profile operations, making it difficult to guarantee reliability of the memory maintenance process.

6 Conclusion

In this paper, we draw inspiration from dialectical philosophy and propose PraMem, which conducts beforehand practice over the lengthy historical sequence to build an experiential memory, thereby assisting for accurate long-horizon behavior prediction. The PraMem iteratively performs existing experience trial, reflective proposal generation, and consensual experience adjustment to maintain pattern experience for presenting user behavioral patterns and bias-alert experience for alerting LLM intrinsic biases. And we design the self-review mechanism, which can filter out reflective proposals without sufficient groundedness and good generalizability, thereby ensuring the overall effectiveness of PraMem. Experiments on various long-horizon behavior prediction tasks demonstrate PraMem is a powerful method with better overall performance than all baselines, and provide some valuable insights of experiential memory. Therefore, this paper offers a promising direction for more powerful memory systems for long-horizon behavior prediction in the future.

REFERENCES

Islem Bouzenia, Premkumar Devanbu, and Michael Pradel. Repairagent: An autonomous, llm-based agent for program repair. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pp. 2188–2200. IEEE, 2025.

Jiawei Chen, Ruoxi Xu, Boxi Cao, Ruotong Pan, Yunfei Zhang, Yifei Hu, Yong Du, Tingting Gao, Yaojie Lu, Yingfei Sun, et al. Towards real-world human behavior simulation: Benchmarking large language models on long-horizon, cross-scenario, heterogeneous behavior traces. *arXiv preprint arXiv:2604.08362*, 2026.

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

Friedrich Engels. *Dialectics of nature*. Wellred Books, 1960.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. Memp: Exploring agent procedural memory. *arXiv preprint arXiv:2508.06433*, 2025.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, et al. A survey of self-evolving agents: On path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 1, 2025.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, et al. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*, 2025.

Jiani Huang, Xingchen Zou, Lianghao Xia, and Qing Li. Mr. rec: Synergizing memory and reasoning for personalized recommendation assistant with llms. *arXiv preprint arXiv:2510.14629*, 2025.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

Bowen Jiang, Zhuoqun Hao, Young Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo Jose Taylor, and Dan Roth. Know me, respond to me: Benchmarking LLMs for dynamic user profiling and personalized responses at scale. In *NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling*, 2025. URL https://openreview.net/forum?id=RheZHR1LSA.

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

Borui Li, Xingcai Zhang, Tianen Liu, Shuai Wang, and Yun Cheng. Digimongpt: An evolvable agent with hierarchical human-like memory for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 6001–6009, 2026.

Jiachun Li, Pengfei Cao, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. MIRAGE: Evaluating and explaining inductive reasoning process in language models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=tZCqSVncRf.

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

Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. In *Proceedings of the ACM Web Conference 2024*, pp. 3497–3508, 2024.

Gang Liu, Fan Yang, Yang Jiao, Alireza Bagheri Garakani, Tian Tong, Yan Gao, and Meng Jiang. Learning attribute as explicit relation for sequential recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 800–811, 2025.

Hongyang Liu, Zhu Sun, Tianjun Wei, Yan Wang, Jiajie Zhu, and Xinghua Qu. Diagnostic-guided dynamic profile optimization for llm-based user simulators in sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15306–15314, 2026.

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

James L McClelland. The place of modeling in cognitive science. *Topics in Cognitive Science*, 1 (1):11–38, 2009.

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

Vladimir M Petrović. Artificial intelligence and virtual worlds–toward human-level ai agents. *IEEE Access*, 6:39976–39988, 2018.

Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2671–2679, 2019.

Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, et al. Lifelong sequential modeling with personalized memorization for user response prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 565–574, 2019.

Yasuaki Sumita, Koh Takeuchi, and Hisashi Kashima. Cognitive biases in large language models: A survey and mitigation experiments. In *Proceedings of the 40th ACM/sigapp symposium on applied computing*, pp. 1009–1011, 2025.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023a.

Lei Wang, Jingsen Zhang, Hao Yang, Zhi-Yuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Hao Sun, Ruihua Song, et al. User behavior simulation with large language model-based agents. *ACM Transactions on Information Systems*, 43(2):1–37, 2025a.

Lu Wang, Di Zhang, Fangkai Yang, Pu Zhao, Jianfeng Liu, Yuefeng Zhan, Hao Sun, Qingwei Lin, Weiwei Deng, Dongmei Zhang, et al. Lettingo: Explore user profile generation for recommendation system. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2985–2995, 2025b.

Qingyue Wang, Liang Ding, Yanan Cao, Zhiliang Tian, Shi Wang, Dacheng Tao, and Li Guo. Recursively summarizing enables long-term dialogue memory in large language models. *arXiv preprint arXiv:2308.15022*, 2023b.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. Recmind: Large language model powered agent for recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 4351–4364, 2024.

Zhiyuan Weng, Guikun Chen, and Wenguan Wang. Do as we do, not as you think: the conformity of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=st77ShxP1K.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. Memocrs: Memory-enhanced sequential conversational recommender systems with large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2585–2595, 2024.

Ke Yang, Zixi Chen, Xuan He, Jize Jiang, Michel Galley, Chenglong Wang, Jianfeng Gao, Jiawei Han, and ChengXiang Zhai. Plugmem: A task-agnostic plugin memory module for llm agents. *arXiv preprint arXiv:2603.03296*, 2026.

Mao Zedong. On contradiction. *Chinese Studies in Philosophy*, 19(2):20–82, 1987.

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. On generative agents in recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in Information Retrieval*, pp. 1807–1817, 2024.

Guibin Zhang, Haotian Ren, Chong Zhan, Zhenhong Zhou, Junhao Wang, He Zhu, Wangchunshu Zhou, and Shuicheng Yan. Memevolve: Meta-evolution of agent memory systems. *arXiv preprint arXiv:2512.18746*, 2025.

Weizhi Zhang, Xiaokai Wei, Wei-Chieh Huang, Zheng Hui, Chen Wang, Michelle Gong, and Philip S Yu. Memorycd: Benchmarking long-context user memory of llm agents for lifelong cross-domain personalization. *arXiv preprint arXiv:2603.25973*, 2026a.

Yi Zhang, Yiwen Zhang, Yu Wang, Tong Chen, and Hongzhi Yin. Proex: A unified framework leveraging large language model with profile extrapolation for recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1940–1951, 2026b.

Boyuan Zheng, Michael Y Fatemi, Xiaolong Jin, Zora Zhiruo Wang, Apurva Gandhi, Yueqi Song, Yu Gu, Jayanth Srinivasa, Gaowen Liu, Graham Neubig, et al. Skillweaver: Web agents can self-improve by discovering and honing skills. *arXiv preprint arXiv:2504.07079*, 2025.

Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. Synapse: Trajectory-as-exemplar prompting with memory for computer control. *arXiv preprint arXiv:2306.07863*, 2023.

Zhi Zheng, Wenshuo Chao, Zhaopeng Qiu, Hengshu Zhu, and Hui Xiong. Harnessing large language models for text-rich sequential recommendation. In *Proceedings of the ACM Web Conference 2024*, pp. 3207–3216, 2024.

Jialiang Zhu, Gongrui Zhang, Xiaolong Ma, Lin Xu, Miaosen Zhang, Ruiqi Yang, Song Wang, Kai Qiu, Zhirong Wu, Qi Dai, et al. Re-trac: Recursive trajectory compression for deep search agents. *arXiv preprint arXiv:2602.02486*, 2026.

A LIMITATIONS.

A main limitation of this paper lies in the memory construction cost of PraMem. Compared with conventional memory management methods based on information extraction, aggregation, and retrieval, the iterative practice process of PraMem inevitably requires more time to build the experiential memory for better performance. Although this process is conducted offline and thus does not affect the efficiency of the actual prediction, it remains a non-trivial limitation that calls for future investigation, for instance through more efficient practice scheduling or parallelized trial-and-reflection strategies.

Prompt for Existing Experience Trial

```
prompt_for_predict = """{practice_item_simulated_input}

---

## When completing the task above, you may refer to and use the experience memory

The experience memory is divided into two categories:

【Pattern Experience (User Behavioral Patterns)】
Records the preferences and behavioral habits of this user; the subject being described is "this user."

【Calibration Experience (Cognitive Calibration and Reminders)】
Records the error tendencies, overlooked factors, or points requiring special attention that the model may encounter when reasoning about this user's behavior; the subject being described is "I (the model) when reasoning."

---

{exp_memory_text}

---

Based on the experience memory above and the historical behavior sequence, make a prediction for each practice task.

Before predicting each task, first complete the following two reasoning steps:
1. List the Pattern Experience entries relevant to this task and explain this user's behavioral tendencies in this type of scenario
2. List the Calibration Experience entries relevant to this task and explain what I need to pay attention to when reasoning

Then provide the prediction result based on the reasoning above.

Answer each task one by one in the given order, and output each task separately."""
```

Figure 6: Prompt for existing experience trial.

B AI USE STATEMENT

No generative AI tools were used at any stage of this work. All research questions, hypotheses, theoretical or conceptual frameworks, mathematical claims and proofs, methodologies, experiments, data generation and processing, method implementations, analyses, and interpretations were developed and conducted solely by the authors. Generative AI was also not used for literature search or review, brainstorming, translation, coding, figure or artifact creation, transcription, paper drafting or editing, reference formatting, or suggesting the paper's structure, title, or keywords. This decision was made to maintain direct authorial control and ensure the rigor and integrity of the research. The authors take full responsibility for all content, claims, code, data, analyses, and other artifacts presented in this work.

C PROMPT FOR EXISTING EXPERIENCE TRIAL

This prompt is used in the existing experience trial stage of PraMem, where the LLM performs beforehand practice on constructed practice samples from the historical sequence. Given the current experiential memory, the LLM is required to predict the action for each practice task based on both the historical behavior sequence and the accumulated experience. To make the use of experiential memory explicit and controllable, the prompt asks the LLM to separately identify relevant pattern experience, which captures this user's behavioral patterns, and relevant deficiency experience, which alerts the model to its own possible intrinsic deficiencies when reasoning about this user. This explicit deep-thinking process helps expose whether the current memory can effectively support prediction and provides necessary reasoning traces for subsequent reflection. The prompt is in Figure 6.

D PROMPT FOR REFLECTIVE PROPOSAL GENERATION

This prompt is used in the reflective proposal generation stage of PraMem, where the LLM reflects on the just-completed practice round by comparing its predictions with the ground-truth labels. Its purpose is to transform trial-and-error outcomes into candidate revisions for experiential memory, so that the model can gradually induce reliable behavioral patterns and identify its own deficiencies for the current user. The prompt requires the LLM to analyze both correct and incorrect predictions, since correct predictions may reinforce useful experience or expose accidental correctness, while incorrect predictions can reveal missing, inaccurate, or conflicting experience. It also explicitly distinguishes pattern experience, which describes this user's behavioral patterns, from deficiency

Prompt for Reflective Proposal Generation

```
prompt_for_reflect = """# Background
We are predicting the behavior of a user based on a long sequence of historical behavior. This user has hundreds of historical behavior records, each containing a scenario description and the user's response. The ultimate goal is to predict the user's behavior in new scenarios.

To improve prediction accuracy, we use a Self-Practice method: simulated practice questions with ground-truth answers are constructed from the user's historical behavior sequence, allowing the model to practice, reflect, and generalize repeatedly, and to accumulate two categories of experience memory before making formal predictions:
- Pattern Experience (User Behavioral Patterns): generalizations about this user's preferences and behavioral habits
- Calibration Experience (Cognitive Bias Calibration): error tendencies and points requiring attention when the model reasons about this user's behavior

Each round of practice consists of three steps: ① predict the answers to the practice questions based on the current experience memory; ② reflect by comparing the predictions with the ground-truth answers and generate experience revision proposals; ③ review the proposal pool every few rounds and consolidate viewpoints supported by multiple proposals into formal experience. To ensure proposal quality, each proposal must also pass two self-review tests before entering the proposal pool: a perturbation sensitivity test (to ensure that the proposal genuinely depends on historical data) and a generalizability test (to ensure that the proposal is not applicable only to one specific scenario).

Your current task is 【Step ②: Reflection and Proposals】, which means deeply reflecting on this round of predictions by comparing them with the ground-truth answers and proposing revisions to the experience memory.

---

You are a rigorous quality analysis expert specializing in behavioral prediction. Your task is to deeply reflect on and summarize the prediction tasks just completed, and to propose revisions to the experience memory used when completing the tasks.

# Experience Memory Description

The experience memory is divided into two categories:

【Pattern Experience (User Behavioral Patterns)】
Records the preferences and behavioral habits of this user; the subject being described is "this user."
Correct example: "This user prefers livestream content featuring outdoor natural settings"
Incorrect example: "When making predictions, check whether natural settings are present" (This is a reminder for the model, not a user characteristic)

【Calibration Experience (Cognitive Calibration and Reminders)】
Records the error tendencies, overlooked factors, or points requiring special attention that the model may encounter when reasoning about this user's behavior; the subject being described is "I (the model) when reasoning."
Correct example: "I tend to overestimate how frequently this user interacts with product-promotion content; the actual level of interaction is relatively low"
Correct example: "When predicting this user's response to emotional content, I need to pay additional attention to the time-of-day factor"
Incorrect example: "This user dislikes product-promotion content" (This is a user characteristic and should be placed under Pattern Experience)

---

# Review of This Practice Round

{exp_memory_text}

---

## Original Practice Question Input
{practice_item_simulated_input}

---

## Model's Prediction Results
{predict_response}

---

## Ground-Truth Answers
{practice_item_simulated_label}

---

# In-Depth Reflection and Summary

## Step 1: Evaluate the Prediction for Each Task
For each task, determine whether the prediction was correct or showed a clear deviation, and label it as "Correct" or "Incorrect/Deviated."

## Step 2: Analyze the Experience for Each Task
**Analyze every task, regardless of whether the prediction was correct.**

- For tasks predicted **correctly**, summarize what led to success, focusing on:
  -Did this success reveal any new user pattern or characteristic of the model's reasoning?
  -Which experiences directly supported the correct reasoning? These may be reinforced
  -Was the prediction correct merely "by coincidence" (the experience-based logic was flawed, but the result happened to be correct)? If so, the experience still needs to be revised
- For tasks predicted **incorrectly**, analyze the root cause, focusing on:
  - Was a particular Pattern Experience inaccurate or incomplete?
  - Was a particular Calibration Experience missing, causing an omission or bias during reasoning?
  - Did an existing experience conflict with this scenario or fail to apply to it?

## Step 3: Generate Revision Proposals
Proposals may be generated for all tasks. Before generating a proposal, check whether it meets all of the following quality requirements:

1. 【Correct Classification】Clearly determine whether the proposal belongs to Pattern Experience or Calibration Experience; do not confuse the two
2. 【Usability】The experience must not depend on external information unavailable to the model during reasoning (such as user identity data or real-time live-comment statistics)
3. 【Assessability】The conditions in the experience must be directly identifiable from the input's content description; avoid details that require inference to confirm
4. 【Avoid False Precision】Do not use specific numbers without statistical support (such as "a correction coefficient of 0.1" or "extend by 300%"); use degree-based wording instead (such as "extend significantly" or "increase slightly")
5. 【Generalizability】The experience should apply to similar scenarios; do not directly turn a one-off special observation into an experience

---

Please output the reflection results in the following format:

## Prediction Evaluation for Each Task
(Explain whether each task's prediction was correct, labeling it as "Correct" or "Incorrect/Deviated")

## Experience Analysis and Proposals for Each Task
(Analyze each task and list the corresponding proposals; if no new proposal can be generated for a task, state "No new proposal")
Each proposal must appear on a separate line as a single line of text beginning with "【Proposal】", in the following format:
【Proposal】Brief description of the triggering scenario | Proposal type: Add Pattern/Add Calibration/Modify existing experience/Remove existing experience | Specific content

(If no new proposals are generated for any task in this round, write "No revision proposals for this round")"""
```

Figure 7: Prompt for reflective proposal generation.

experience, which records the model's possible mistakes or overlooked aspects, ensuring that reflective proposals are assigned to the proper memory category. The prompt is shown in Figure 7.

Prompt for Consensual Experience Adjustment

```
prompt_for_review = """# Background
We are conducting behavior prediction based on a user's long historical behavior sequence. This user has hundreds of historical behavior records, each containing a scenario description and the user's response. The ultimate goal is to predict this user's behavior in new scenarios.

To improve prediction accuracy, we use a Self-Practice method: simulated practice questions with ground-truth answers are constructed from the user's historical behavior sequence, allowing the model to repeatedly practice, reflect, and summarize before formal prediction, accumulating two types of experiential memory:
- Pattern Experience: summaries of this user's preferences and behavioral habits
- Calibration Experience: cognitive-bias calibration about the model's own error tendencies and points requiring attention when reasoning about this user's behavior

Each practice round contains three steps: ① predict the practice questions based on the current experiential memory; ② reflect against the ground-truth answers and generate experience-revision proposals; ③ every few rounds, review the proposal pool and consolidate viewpoints supported by multiple proposals into formal experience. To ensure proposal quality, each proposal must also pass two self-audit checks before entering the proposal pool: the perturbation-sensitivity check, which ensures the proposal truly depends on historical data, and the generalizability check, which ensures the proposal is not only applicable to one specific scenario.

Your current task is 【Step ③: Experience Review】, namely to prudently update experiential memory based on the proposal pool. Only viewpoints supported by multiple proposals should be incorporated.

---

You are an experience-management expert. Your task is to prudently update the current experiential memory based on the accumulated revision proposals.

## Explanation of Experiential Memory

Experiential memory is divided into two categories:

【Pattern Experience】
Records this user's own preferences and behavioral habits. The described subject is "this user."

【Calibration Experience】
Records the model's tendencies toward mistakes, overlooked points, or aspects requiring special attention when reasoning about this user's behavior. The described subject is "I (the model) when reasoning."

---

## Current Experiential Memory State

{exp_memory_text}

### Proposal Pool to Be Reviewed
{proposal_text}

---

## Review Task

### Step 1: Group the Proposals
Group all proposals in the proposal pool by content similarity, and determine which proposals express the same viewpoint.

### Step 2: Process Each Proposal Group According to the Adoption Rules
Strictly follow the rules below; do not skip any:

**Rules for adding or modifying experience:**
- If a viewpoint is supported by 3 or more proposals → it may be incorporated into experiential memory or used to modify existing experience.
- If a viewpoint is supported by only 1–2 proposals → it must remain in the proposal pool and must not be incorporated into experiential memory.

**Rules for deleting or downgrading existing experience:**
- If an existing experience is questioned by 3 or more proposals → modify that experience or downgrade it to the proposal pool.
- If an existing experience is questioned by only 1–2 proposals → keep it unchanged.

**Note: The proposal pool is not a to-do list, but a set of candidate experiences under observation. Under normal circumstances, most proposals should still remain in the proposal pool after each review.**

### Step 3: Check Experience Quality
Before incorporating any proposal into experiential memory, check whether each experience satisfies the following requirements:
1. 【Correct Classification】Pattern Experience describes the user; Calibration Experience describes points to note during model reasoning. Do not confuse the two.
2. 【Usability】It must not rely on external information unavailable during model reasoning.
3. 【Judgability】Its conditions can be directly determined from the input content.
4. 【Avoid False Precision】Do not use specific numbers without statistical support; use degree words instead.
5. 【Generalizability】It can apply to similar scenarios and is not a one-off special observation.

---

## Output Requirements

Please output strictly according to the following two steps. Do not merge or skip any step.

### Step 1: Review Decision Explanation
First, explain this round's review decisions in natural language:
- Which proposal groups are adopted (indicate the corresponding proposal numbers)
- Which existing experiences are modified or downgraded (indicate the specific entries)
- The remaining proposals stay in the proposal pool and do not need to be explained one by one.

### Step 2: Output the Complete Experiential Memory JSON
Output JSON strictly according to the following rules:
- user_exp and model_exp: output the complete experience lists after review.
- proposal: must contain the complete original text of the following two parts, copied item by item without truncation, summarization, or ellipses:
  (1) all proposals judged as "retained" in Step 1, copied verbatim from the original text
  (2) any existing experiences downgraded in Step 1

```json
{{
  "user_exp": [...],
  "model_exp": [...],
  "proposal": [...]
}}
```"""
```

Figure 8: Prompt for consensual experience adjustment.

E PROMPT FOR CONSENSUAL EXPERIENCE ADJUSTMENT

This prompt is used in the consensual experience adjustment stage of PraMem, where the accumulated proposal pool is periodically reviewed to update the experiential memory. Its role is to prevent unstable memory changes caused by occasional behaviors or noisy reflections, which is essential for building reliable time-evolving experience from long historical sequences. The prompt asks the LLM to group similar proposals and adopt only viewpoints supported by multiple proposals, while

weakly supported proposals remain in the proposal pool for further observation. It also applies quality checks before incorporation, ensuring that the updated pattern experience and deficiency experience remain usable, distinguishable, and generalizable. The prompt is shown in Figure 8.

F Prompt for Groundedness Review

This prompt is used in the self-review mechanism to conduct the perturbation-based groundedness check for reflective proposals. In PraMem, a reliable proposal should be genuinely derived from the historical sequence rather than being a generic or hallucinated statement produced by the LLM. To test this, the prompt first asks the LLM to design strong perturbation plans that substantially change the preference signals conveyed by the historical behavior sequence. Then, based on these perturbation plans, the LLM judges whether each original proposal would still hold after the historical sequence is perturbed; proposals that remain valid under strong perturbation are regarded as insufficiently grounded and are filtered out. The prompt is shown in Figure 9.

G Prompt for Generalizability Review

This prompt is used in the self-review mechanism to conduct the generalizability check for reflective proposals. In PraMem, a useful proposal should not merely memorize one practice sample, but should capture experience that can transfer to similar prediction scenarios. To test this, the prompt first generates several virtual scenes that share the same broad type and format as the real prediction scene but differ in key entities or details. Then, the LLM judges whether each proposal can uniquely identify the original real scene among these candidate scenes; if it can, the proposal is considered overly specific to the current practice sample and is filtered out. The prompt is shown in Figure 10.

H Case Study

To more intuitively demonstrate the essential advantages of the experiential memory constructed by PraMem over traditional memory management methods, we conduct a case study that compares the memory content maintained for the same user under PraMem and the representative baseline method MemOS, as shown in Figure 11.

In this case, the memory distilled by PraMem consists of two parts: on one hand, the pattern experience characterizes the user's behavioral regularities in a highly condensed manner, such as "the user is more inclined to follow authors whose tags contain social or emotional elements" and "the viewing duration is jointly determined by video content type, title keywords, and video length"; on the other hand, the bias-alert experience explicitly flags the cognitive biases that the LLM tends to fall into when facing this user, such as "alert against over-generalizing from frequent brief-dwell records in the history into extremely low engagement predictions" and "alert against the training-induced prejudice that treats following an author as a rare, need-driven action by default". In contrast, the memory maintained by MemOS is primarily a factual recount of the user's historical behaviors, recording concrete entries such as "on 2025-09-28 at 20:16:18, the user watched a 57-second video about gypsum board partition walls for 3 seconds", with each entry confined to an objective description of a single behavior.

From this, it can be seen that the memories constructed by PraMem and the baseline method differ fundamentally in nature: the baseline's memory is a record of facts, and even after compression and retrieval, its content still remains at the level of "what has happened", requiring the LLM to induce behavioral patterns on the fly from fragmented records during the prediction stage, while being powerless against the model's intrinsic cognitive biases; in contrast, PraMem's memory is the experience distilled through iterative practice, which not only explicitly presents the user's latent behavioral patterns in a high-level abstract manner, but also pre-alerts the LLM against the systematic errors it tends to commit on this user, thereby directly acting on the prediction process and genuinely resolving the two core challenges of long-horizon behavior prediction.

Prompt for Groundedness Review

```
prompt_for_generate_perturbations = """# Background
We are performing behavior prediction for a user with a long historical behavior sequence. This user has hundreds of historical behavior records, each containing a scenario description and the user's response. The ultimate goal is to predict the user's behavior in a new scenario.

To improve prediction accuracy, we adopt a Self-Practice method: we construct simulated practice questions with ground-truth answers from the user's historical behavior sequence, allowing the model to practice, reflect, and summarize repeatedly. Before the formal prediction, the model accumulates two types of experiential memory:
- Pattern Experience: summaries of the user's preferences and behavioral habits
- Calibration Experience: corrections of the model's own cognitive biases, including error tendencies and points requiring attention when reasoning about this user's behavior

Each practice round contains three steps: ① make a prediction for the practice question based on the current experiential memory; ② compare with the ground-truth answer and reflect, generating proposed experience revisions; ③ every several rounds, review the proposal pool and consolidate views supported by multiple proposals into formal experience. To ensure proposal quality, each proposal must also pass two self-audit checks before entering the proposal pool: a perturbation sensitivity check, which ensures the proposal truly depends on historical data, and a generalizability check, which ensures the proposal does not apply only to one specific scenario.

Your current task is [Self-Audit · Perturbation Sensitivity Check Step 1: Generate Perturbation Plans], namely to design strong perturbation plans for the historical behavior sequence, used to test whether a proposal truly depends on historical data.

---

You are an expert in behavioral data perturbation. Your task is to design strong perturbation plans for the given user's historical behavior trajectory, in order to test whether a certain experience proposal truly depends on historical data.

# Perturbation Principles

The goal of perturbation is to substantially modify the [response] field in historical behaviors, so that the perturbed historical sequence conveys user preference signals that are completely different from the original sequence:
- **The perturbation must be aggressive enough**: Do not make only minor adjustments. You must make substantive reversals or subversions, so that the user profile before and after perturbation differs significantly.
- **The perturbation must cover multiple behaviors**: Do not modify only one item. Each perturbation should cover all or the vast majority of behavior entries in the historical sequence, forming a systematic change in preference signals.
- **The perturbation direction must be clear**: The perturbation should have a clear directional theme, for example:
  - Direction A: Change all positive/active responses, such as clicking, purchasing, liking, and converting, into negative/passive responses, such as swiping away, ignoring, or viewing without interaction.
  - Direction B: Change all high-intensity interactions, such as multiple likes, favorites, and comments, into zero interaction, such as only being exposed without clicking.
  - Direction C: Change all purchase/conversion behaviors into abandonment behaviors, and change all likes into reports or "not interested" feedback.

# Perturbation Strength Self-Check
After designing each perturbation plan, ask yourself:
- If a person only looked at the perturbed historical sequence, would their judgment of this user's preferences be significantly different from what they would infer from the original sequence?
- If the answer is "not much difference," the perturbation is not strong enough and needs to be strengthened.

# Original Historical Behavior Trajectory

{history_text}

---

# Output Requirements

Please design {num_perturbations} strong perturbation plans. For each plan, you only need to describe the perturbation idea; you do not need to output the full perturbed sequence. Output in JSON format as follows:

```json
[
  {{
    "perturbation_desc": "(describe the directional theme of this perturbation)",
    "perturbation_details": "(combine with the specific content of the original sequence to explain which types of behavioral responses will be modified and how, including before-and-after comparison examples)"
  }},
  ...
]```
"""



prompt_for_check_perturbation = """# Background
We are performing behavior prediction for a user with a long historical behavior sequence. This user has hundreds of historical behavior records, each containing a scenario description and the user's response. The ultimate goal is to predict the user's behavior in a new scenario.

To improve prediction accuracy, we adopt a Self-Practice method: we construct simulated practice questions with ground-truth answers from the user's historical behavior sequence, allowing the model to practice, reflect, and summarize repeatedly. Before the formal prediction, the model accumulates two types of experiential memory:
- Pattern Experience: summaries of the user's preferences and behavioral habits
- Calibration Experience: corrections of the model's own cognitive biases, including error tendencies and points requiring attention when reasoning about this user's behavior

Each practice round contains three steps: ① make a prediction for the practice question based on the current experiential memory; ② compare with the ground-truth answer and reflect, generating proposed experience revisions; ③ every several rounds, review the proposal pool and consolidate views supported by multiple proposals into formal experience. To ensure proposal quality, each proposal must also pass two self-audit checks before entering the proposal pool: a perturbation sensitivity check, which ensures the proposal truly depends on historical data, and a generalizability check, which ensures the proposal does not apply only to one specific scenario.

Your current task is [Self-Audit · Perturbation Sensitivity Check Step 2: Judge Sensitivity], namely to infer, based on the perturbation plan, whether the original proposal would still hold if the historical sequence were perturbed according to that plan, thereby filtering out proposals hallucinated by the model.

---

You are an expert in auditing the quality of experience proposals. Your task is to determine whether a batch of experience proposals truly depends on the user's historical behavior data, rather than being subjective hallucinations by the model.

# Audit Logic

We have designed a purposeful perturbation plan for the user's historical behavior trajectory. The perturbation is not actually executed; only the perturbation description is provided. You need to combine the original historical sequence and the perturbation plan to infer whether each proposal would still hold if the perturbation were executed:
- If the proposal **no longer holds** after the perturbation is executed, this indicates that the proposal truly depends on the specific content of the historical data and is a well-grounded conclusion.
- If the proposal **still holds** after the perturbation is executed, this indicates that the proposal is unrelated to the specific content of the historical data and is a hallucinated conclusion by the model.

# Audit Materials

## Original Historical Behavior Sequence
{history_text}

## Current Perturbation Plan
**Perturbation Direction**: {perturbation_desc}
**Perturbation Details**: {perturbation_details}

## All Original Proposals to Be Audited
{original_proposals_text}

---

# Audit Steps

## Step 1: Understand the Perturbation Effect
Combine the original historical sequence and the perturbation plan to mentally construct the overall user profile after perturbation: what distinctly different preference characteristics would the perturbed user exhibit?

## Step 2: Audit Each Proposal One by One
For each original proposal to be audited, determine whether its conclusion was derived from specific behavioral features in the original sequence, or whether it is a generalized assertion unrelated to the content of the historical data. If the perturbed user profile contradicts the pattern described by the proposal, then the proposal is sensitive and well-grounded. If the perturbed user profile remains compatible with the proposal, then the proposal is insensitive and hallucinated.

## Step 3: Output the Conclusion
Output a conclusion for each proposal in the following strict format:

```json
[
  {{"proposal_idx": 0, "is_sensitive": true, "reason": "(briefly explain which specific features in the original sequence supported this proposal, and why the proposal no longer holds after these features disappear due to perturbation)"}},
  {{"proposal_idx": 1, "is_sensitive": false, "reason": "(briefly explain why this proposal is unrelated to the specific content of the historical data and would still hold after perturbation)"}},
  ...
]```

Note: proposal_idx corresponds one-to-one with the numbering of the proposals to be audited, starting from 0."""
```

Figure 9: Prompt for groundedness review.

Prompt for Generalizability Review

```
prompt_for_generate_virtual_scenes = """# Background
We are performing behavior prediction for a user with a long historical behavior sequence. This user has hundreds of historical behavior records, each containing a scenario description and the user's response. The ultimate goal is to predict the user's behavior in a new scenario.

To improve prediction accuracy, we adopt a Self-Practice method: we construct simulated practice questions with ground-truth answers from the user's historical behavior sequence, allowing the model to practice, reflect, and summarize repeatedly. Before the formal prediction, the model accumulates two types of experiential memory:
- Pattern Experience: summaries of the user's preferences and behavioral habits
- Calibration Experience: corrections of the model's own cognitive biases, including error tendencies and points requiring attention when reasoning about this user's behavior

Each practice round contains three steps: ① make a prediction for the practice question based on the current experiential memory; ② compare with the ground-truth answer and reflect, generating proposed experience revisions; ③ every several rounds, review the proposal pool and consolidate views supported by multiple proposals into formal experience. To ensure proposal quality, each proposal must also pass two self-audit checks before entering the proposal pool: a perturbation sensitivity check, which ensures the proposal truly depends on historical data, and a generalizability check, which ensures the proposal does not apply only to one specific scenario.

Your current task is [Self-Audit · Generalizability Check Step 1: Generate Virtual Scenarios], namely to generate similar virtual scenarios based on a real scenario, used to test whether a proposal overly depends on a specific scenario.

---

You are a scenario generation expert. Your task is to generate several virtual scenarios that are of the same type as, but not exactly the same as, a real target prediction scenario, in order to test the generalizability of a certain experience proposal.

# Generation Principles

- The virtual scenarios should belong to the same broad category as the real scenario, such as mall shopping scenarios, advertisement recommendation scenarios, etc.
- The virtual scenarios need to identify the key entities in the real scenario, such as product names, categories, industries, prediction questions, etc., and replace them with entities of the same type but different from the original ones.
- The replaced virtual scenarios should remain consistent with the real scenario in structure and format.
- The virtual scenarios should be sufficiently different from one another and must not be too similar to each other.
- Do not modify fixed-format parts of the scenario, such as "The scenario details are as follows:" or question templates like "Will this user add this product to the shopping cart?"

# Real Scenario

{real_scene_text}

---

# Output Requirements

Please generate {num_virtual_scenes} virtual scenarios and output them in JSON format as follows:

```json
[
  "(full text of virtual scenario 1, with exactly the same format as the real scenario)",
  "(full text of virtual scenario 2)",
  ...
]```

Note: Each virtual scenario must be a complete scenario text, including the scenario details and the prediction question."""



prompt_for_check_generalization = """# Background
We are performing behavior prediction for a user with a long historical behavior sequence. This user has hundreds of historical behavior records, each containing a scenario description and the user's response. The ultimate goal is to predict the user's behavior in a new scenario.

To improve prediction accuracy, we adopt a Self-Practice method: we construct simulated practice questions with ground-truth answers from the user's historical behavior sequence, allowing the model to practice, reflect, and summarize repeatedly. Before the formal prediction, the model accumulates two types of experiential memory:
- Pattern Experience: summaries of the user's preferences and behavioral habits
- Calibration Experience: corrections of the model's own cognitive biases, including error tendencies and points requiring attention when reasoning about this user's behavior

Each practice round contains three steps: ① make a prediction for the practice question based on the current experiential memory; ② compare with the ground-truth answer and reflect, generating proposed experience revisions; ③ every several rounds, review the proposal pool and consolidate views supported by multiple proposals into formal experience. To ensure proposal quality, each proposal must also pass two self-audit checks before entering the proposal pool: a perturbation sensitivity check, which ensures the proposal truly depends on historical data, and a generalizability check, which ensures the proposal does not apply only to one specific scenario.

Your current task is [Self-Audit · Generalizability Check Step 2: Judge Generalizability], namely to determine whether each proposal is too specific and only applicable to the current specific scenario, thereby filtering out proposals with insufficient generalizability.

---

You are an expert in auditing the generalizability of experience proposals. Your task is to determine whether a batch of experience proposals has generalizability.

# Audit Logic

A good experience proposal should have generalizability and be applicable to multiple scenarios of the same type, rather than targeting only one specific scenario.
- If a proposal is described too specifically and is highly bound to a particular scenario, then the original scenario corresponding to it can be precisely identified based on that proposal.
- If a proposal has good generalizability and describes a general pattern, then it is impossible to determine which specific scenario it corresponds to based solely on the proposal content.

# Audit Materials

## Candidate Scenarios (A is the real scenario, while the others are similar virtual scenarios)
{options_text}

## All Experience Proposals to Be Audited
{proposals_text}

---

# Audit Steps

## Step 1: Understand the Differences Among Candidate Scenarios
Identify the key distinguishing entities or features in each candidate scenario.

## Step 2: Audit Each Proposal One by One
For each proposal, determine whether the real scenario A can be precisely identified based solely on the content of the proposal, i.e., whether the proposal is too specific.
- If A can be precisely identified, this indicates poor generalizability, and the proposal fails the check.
- If A cannot be precisely identified, this indicates good generalizability, and the proposal passes the check.

## Step 3: Output the Conclusion
Output a conclusion for each proposal in the following strict format:

```json
[
  {{"proposal_idx": 0, "generalization_passed": true, "reason": "(brief explanation)"}},
  {{"proposal_idx": 1, "generalization_passed": false, "reason": "(brief explanation)"}},
  ...
]```

Note: proposal_idx corresponds one-to-one with the numbering of the proposals to be audited, starting from 0. The real scenario is fixed as option A; the model does not need to guess."""
```

Figure 10: Prompt for generalizability review.

Case Study

PraMem

Pattern Experience

- The user is more inclined to follow authors whose tags contain social or emotional elements.
- This user's viewing duration is jointly influenced by video content type, title keywords, and video length: light-entertainment, humorous, or lifestyle short videos typically receive brief dwell time; when a video contains keywords related to family, challenges, competition, or emotional relationships, and ratio is significantly higher than the user's historical average, the user tends to extend watching, often viewing the video in full; for short videos no longer than 2 minutes with a full-play ratio at least twice the historical average, the user exhibits a high probability of complete viewing; overall, combination of content type and video length determines.

…

Bias-alert Experience

- Alert against over-generalizing from the majority of brief-dwell records in the user's history: the LLM tends to default to predicting extremely low engagement (likes, comments, shares, viewing duration) based on frequent short-dwell patterns, while overlooking that utility/emotional-conflict content, …
- Alert against the training-induced prejudice that treats 'follow author' as a rare, need-driven action by default: in entertainment-driven scenarios with high full-play ratios, high author follower counts, or strong social/emotional cues, the follow probability should be explicitly re-evaluated rather than dismissed…

…

MemOS

- On 2025-09-28 at 20:16:18, the user watched a video titled '#RenovationSite #RenovationSharing #RenovationDiary Quick Assembly of Gypsum Board Partition Walls,' posted by an author with 242K followers on 2025-08-15 09:53:49. The video is 57 seconds long and received 24,069 impressions, 23,006 plays, 6,504 full plays, 116 likes, 2 comments, 8 shares, 40 collections, and 4 downloads. The user watched for 3 seconds.
- On 2025-09-25 at 20:36, the user browsed a video on the platform about carpentry scaffolding and on-site footage. The creator has 36K followers, and the video documented a renovation process with a duration of about 1 minute 8 seconds. The video performed well in plays and interactions on 2025-08-24, and the user recorded a viewing duration of 6 seconds.
- On 2025-09-19 at 00:37:35, the user watched a video titled 'Coconut Shell Powder Holding-Breath Challenge, All Chaos #LaughEveryTime #HoldingBreathChallenge #CoconutShellPowder #ForgiveMyUnkindLaughter,' 2 minutes 5 seconds long, posted by an author with 1.152M followers on 2025-04-05 15:47:31, receiving 7,014 impressions, 4,006 plays, 641 full plays, 31 likes, 3 shares, 4 collections, and 1 download. The user watched for 2 minutes 6 seconds, completing the entire video.
- On 2025-09-20 at 20:54, the user recorded behavior 25—watching a video titled '#SummerCoolingPlan.' The author has 368K followers; the video is 10 minutes 41 seconds long with impressions, plays, full plays, likes, comments, and shares of 41,420, etc. The user's response was 3 seconds and included zooming in on the video.

…

Figure 11: A case study showing that PraMem distills abstract behavioral patterns and bias alerts through practice, whereas the baseline MemOS merely records factual historical behaviors.