MoRE: A Mixture of Reflectors Framework for Large Language Model-Based Sequential Recommendation

Weicong Qin*
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
qwc@ruc.edu.cn

Yi Xu*
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
yixu00@ruc.edu.cn

Weijie Yu†[†]
School of Information Technology and Management, University of International Business and Economics
Beijing, China
yu@uibe.edu.cn

Chenglei Shen
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
chengleishen9@ruc.edu.cn

Xiao Zhang
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
zhangx89@ruc.edu.cn

Ming He
AI Lab at Lenovo Research, Lenovo Group Limited
Beijing, China
heming01@foxmail.com

Jianping Fan
AI Lab at Lenovo Research, Lenovo Group Limited
Beijing, China
jfan1@lenovo.com

Jun Xu
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
junxu@ruc.edu.cn

Abstract

Large language models (LLMs) have emerged as a cutting-edge approach in sequential recommendation, leveraging historical interactions to model dynamic user preferences. Current methods mainly focus on learning processed recommendation data in the form of sequence-to-sequence text. While effective, they exhibit three key limitations: 1) failing to decouple *intra-user* explicit features (e.g., product titles) from implicit behavioral patterns (e.g., brand loyalty) within interaction histories; 2) underutilizing *cross-user* collaborative filtering (CF) signals; and 3) relying on inefficient reflection update strategies. To address this, We propose MoRE (Mixture of REflectors), which introduces three perspective-aware offline reflection processes to address these gaps. This decomposition directly resolves Challenges 1 (explicit/implicit ambiguity) and 2 (CF underutilization). Furthermore, MoRE's meta-reflector employs a self-improving strategy and a dynamic selection mechanism (Challenge 3) to adapt to evolving user preferences. First, two *intra-user* reflectors decouple explicit and implicit patterns from a user's interaction sequence, mimicking traditional recommender systems' ability to distinguish surface-level and latent preferences. A third *cross-user* reflector captures CF signals by analyzing user similarity patterns from multiple users' interactions. To optimize reflection quality, MoRE's meta-reflector employs a offline self-improving strategy that evaluates reflection impacts through comparisons of presence/absence and iterative refinement of old/new versions, with a online contextual bandit mechanism dynamically selecting the optimal perspective for recommendation for each user. Experiments on three benchmarks show MoRE outperforms both traditional recommenders and LLM-based methods with minimal computational overhead, validating its effectiveness in bridging LLMs' semantic understanding with multidimensional recommendation principles. Code: https://github.com/E-qin/MoRE-Rec.

CCS Concepts

• **Information systems → Recommender systems**; **Language models**; *Personalization.*

Keywords

Large Language Model Reflection, Sequential Recommendation

ACM Reference Format:
Weicong Qin, Yi Xu, Weijie Yu, Chenglei Shen, Xiao Zhang, Ming He, Jianping Fan, and Jun Xu. 2025. MoRE: A Mixture of Reflectors Framework for Large Language Model-Based Sequential Recommendation. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems (RecSys '25), September 22–26, 2025, Prague, Czech Republic.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3705328.3748055

*The first two authors contributed equally to this research.
†The corresponding author is Weijie Yu.

1 Introduction

Sequential recommendation (SeqRec), which predicts the next item of interest based on a user's historical interaction sequence, is

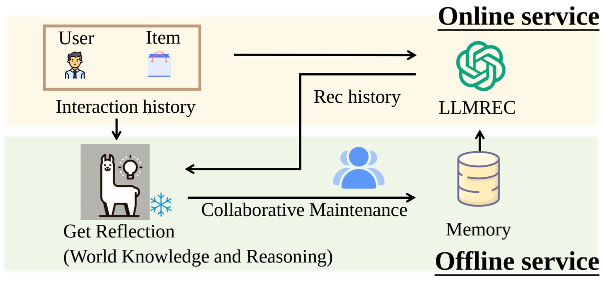


Figure 1: Workflow of reflection-based methods for SeqRec. LLM reflections are generated offline based on interaction and recommendation history. These reflections are then collected as hints to improve online recommendations.

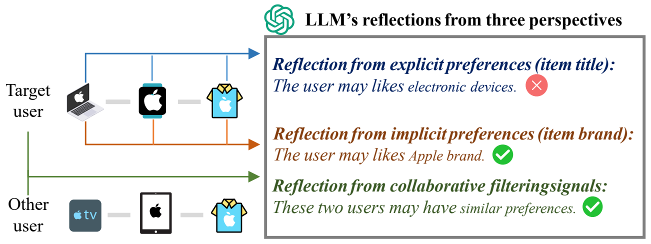

LLM's reflections from three perspectives
Target user
Other user
Reflection from explicit preferences (item title):
The user may likes electronic devices.
Reflection from implicit preferences (item brand):
The user may likes Apple brand.
Reflection from collaborative filtering signals:
These two users may have similar preferences.

Figure 2: Examples of LLM reflections from explicit, implicit, and CF perspectives based on user interactions.

crucial for recommender systems. The key to this task is to capture users' dynamic preferences accurately. Large language models (LLMs) are emerging as promising recommenders (LLMREC) due to their vast world knowledge and excellent reasoning abilities. Two typical approaches in this context are prompt-based methods and fine-tuning methods. Prompt-based methods [4, 11] pre-construct fixed prompts and exploit in-context learning [5] to guide the LLM in reasoning the desired item. Fine-tuning methods [19, 41, 44] inject domain knowledge by fine-tuning an LLM on substantial annotated recommendation data. Although these methods achieve encouraging performance, the former suffers from an inability to optimize the prompts based on user feedback, while the latter requires substantial computational resources.

Recently, [31, 33] propose reflection-based methods where LLMs refine recommendations by analyzing prediction against user interaction (Fig. 1). While effective, they face three key challenges: **(1) Ambiguous separation of explicit features and implicit patterns within one single user's interaction history.** Effectively modeling user preferences in SeqRec requires distinguishing between explicit and implicit signals within user interaction histories. Explicit features are reflected in directly observable item features, such as titles, while implicit patterns are inferred from latent factors in user behavior, such as brand loyalty. Current reflection-based recommendation methods [31, 33] predominantly rely on explicit item features to model user interests, assuming that LLMs can effectively capture preference patterns from surface-level semantics alone. However, this reliance lacks analysis or reflection on the deeper, implicit connections that often drive user behavior. As illustrated in Fig. 2, consider a user who has interacted with several electronic products and later purchases a T-shirt. If the recommender focuses solely on explicit item titles, it struggles to analyze the subtle connection between these interactions and fails to adapt to evolving user interests. In this case, the underlying implicit preference is the brand "Apple", reflecting a deeper pattern that explicit features alone cannot reveal. **(2) Underutilized collaborative filtering (CF) signals across multiple user interaction histories.** CF leverages behavioral similarities across users to detect shared interests (e.g., Apple product fans). Current approaches [31, 33] neglect such signals, limiting their ability to exploit collective behavioral patterns for improved recommendations. **(3) Suboptimal reflection update strategies.** Existing reflection-based methods [31, 33] mainly maintain a fixed reflection pool, lacking personalized dynamic maintenance based on user interactions. This inflexibility prevents them from adapting to users' dynamic preferences.

Facing these challenges, we propose MoRE (Mixture of REflectors) to address the limitations through three coordinated innovations. **First**, on the offline side, we design specialized reflectors to model both *intra-user patterns* and *cross-user patterns*: 1) Two intra-user perspectives disentangle explicit features (e.g., product titles) and implicit behavioral patterns (e.g., brand loyalty) from a single user's interaction history; 2) A cross-user perspective captures collaborative filtering (CF) signals through item-centric analysis of user-item-user similarity relationships. This decomposition directly resolves Challenges 1 (explicit/implicit ambiguity) and 2 (CF underutilization). **Second**, also on the offline side, our self-improving meta-reflector implements a lightweight two-phase update strategy: evaluating reflection quality via LLMREC performance gains (both presence/absence and old/new reflections), then iteratively generating improved reflections using top-performing candidates in an offline loop. This approach overcomes Challenge 3 (suboptimal updates) while maintaining computational efficiency. **Third**, on the online side, the meta-reflector dynamically selects optimal reflections per user through contextual bandit optimization [16]. Experiments demonstrate MoRE's superiority over traditional methods and LLMREC variants (prompt-based, fine-tuning-based, and reflection-based baselines) while maintaining low GPU memory and time costs. In summary, our contributions are as follows:

- We propose a dynamic LLM reflection framework for sequential recommendation to model and learn dynamic user preferences.
- MoRE incorporates three reflectors, each collecting reflections from explicit features, implicit patterns, and CF signals. It also integrates a meta-reflector, which updates the reflections with a refining-and-iteration strategy and selects the appropriate reflection for LLMREC using a contextual bandit algorithm.
- Extensive experiments on three real-world datasets demonstrate that MoRE outperforms state-of-the-art approaches in terms of recommendation performance with minimal GPU memory usage and training time overhead.

2 Related Works

2.1 Sequential Recommendation

SeqRec predicts users' next interests by modeling historical interactions chronologically, with the key challenge in capturing dynamic preferences [23–25, 29, 35–37, 39]. Recent advances [10, 15, 30] employ convolutional and recurrent neural networks, while Li et al. [17, 18], Zhou et al. [45] demonstrate MLP-based approaches achieve competitive performance. He et al. [8, 9], Kang and McAuley [13], Sun et al. [28] utilize transformer-based models for item relevance modeling, achieving SOTA results. Despite progress, existing methods remain limited in holistic preference modeling due to insufficient world knowledge and reasoning capabilities. This has spurred recent exploration of LLM-based approaches for sequential recommendation.

2.2 LLM-based Recommendation

Current approaches to adapting LLMs for SeqRec predominantly fall into two categories: prompt-based methods and fine-tuning methods. Prompt-based methods involve manually constructing fixed prompts to guide LLMs in reasoning toward the desired items. For example, LLM4RS [4] enhances LLM's recommendation capabilities by customizing prompts to align with traditional ranking, while LLMRank [12] leverages LLMs for zero-shot ranking in recommendations through specialized prompting strategies. However, these fixed prompts rely heavily on human expertise and struggle to adapt to diverse users, often resulting in sub-optimal performance [5, 31]. Fine-tuning methods [2, 3, 7, 19, 20, 27, 34, 38], on the other hand, formulate SeqRec as a question answering task and fine-tune accordingly. To incorporate recommendation knowledge, they either add special tokens (e.g., Llara [19], LC-Rec [44]) or insert embeddings as texts into the prompt (e.g., BinLLM [41]). Despite these efforts, a significant gap remains between LLM's pre-training and recommendation-oriented fine-tuning, which necessitates a substantial amount of tuning data to achieve proper alignment. Additionally, [14, 26, 41, 42, 44, 46] attempt to integrate collaborative filtering information into LLMs. While effective, these methods lack an interpretation and analysis of the CF information, which prevents the integration of CF information with the powerful understanding and reasoning capabilities of LLMs, leading to potential suboptimal performance.

Recently, Wang et al. [31] propose Re2LLM, a reflection-based approach that leverages LLMs to reflect on interaction history to enhance future recommendations. However, this approach maintains a static prompt pool without decoupled modeling of explicit and implicit preferences and incorporate CF signals.

3 MoRE: The Proposed Framework

3.1 Problem Formulation

In SeqRec, let $\mathcal{V} = \{v_1, v_2, ..., v_{|\mathcal{V}|}\}$ denotes the item set, $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$ denotes the user set, and $\mathcal{S}_u^t = [v_u^1, v_u^2, ..., v_u^t]$ denotes the interaction sequence in chronological order for user $u \in \mathcal{U}$ up to time step t, where $v_u^t \in \mathcal{V}$ is the item that u has interacted with at time step t. The goal of this task is to predict the item that u will interact with at $t+1$ given $\mathcal{S}_u^t$:

$$\hat{v} = \arg\max_{v \in \mathcal{V}} \Pr(v_u^{t+1} = v \mid \mathcal{S}_u^t). \tag{1}$$

In this study, we focus on the LLM-based recommendation. Following [4, 12], we adopt LLM to directly make the prediction in a ranking fashion as follows:

$$\hat{O}_u^{t+1} = \mathrm{LLM}_{\mathrm{REC}}(P_{\mathrm{REC}}(u, \mathcal{S}_u^t, C_u^{t+1})), \hat{O}_u^{t+1} \subseteq C_u^{t+1}, \tag{2}$$

where C_u^{t+1} and $\hat{O}_u^{t+1}$ respectively denote the candidate and predicted item list for u at step $t+1$; all the parameters are concatenated in the text form; $\mathrm{LLM}_{\mathrm{REC}}$, the LLM for recommendation, is frozen; and P_{REC} denotes the manually crafted recommendation prompt.

More specifically, as illustrated in Fig. 1, we focus on the two-stage reflection-based LLMREC.

In the offline reflection generation stage, given a user u, the interaction sequence $\mathcal{S}_u^{t-1}$ up to time step $t-1$ and the candidate list C_u^t, we utilize LLM to make ranking prediction $\hat{O}_u^t$ at step t:

$$\hat{O}_u^t = \mathrm{LLM}_{\mathrm{REC}}(P_{\mathrm{REC}}(u, \mathcal{S}_u^{t-1}, C_u^t), \hat{O}_u^t \subseteq C_u^t. \tag{3}$$

The LLM then reflects on the predicted list $\hat{O}_u^t$ and the ground-truth item v_u^t to infer u's preferences:

$$Ref_{i,u} = \mathrm{LLM}_{\mathrm{REF}}(P_i(u, \mathcal{S}_u^{t-1}, C_u^t, \hat{O}_u^t, v_u^t)), \tag{4}$$

where $\mathrm{LLM}_{\mathrm{REF}}$ denotes a frozen LLM responsible for generating reflections; the subscript $i \in \{EP, IP, CF\}$ for prompt P_i and reflection $Ref_{i,u}$ corresponds to the "Explicit Preference", "Implicit Preference", and "Collaborative Filtering", which will be detailed in the next section.

In the online recommendation stage, the derived $Ref_{i,u}$ is incorporated into Eq. 2 to enhance future recommendations:

$$\hat{O}_u^{t+1} = \mathrm{LLM}_{\mathrm{REC}}(P_{\mathrm{REC}}(u, \mathcal{S}_u^t, C_u^{t+1}, Ref_{i,u})),\ \hat{O}_u^{t+1} \subseteq C_u^{t+1}. \tag{5}$$

The details of recommendation prompt P_{REC} are as follows:

Recommendation prompt P_{REC}

You are a recommender to recommend items for a specific user. The user interacted with items in the following order: $\langle S_u \rangle$. Reflections on the past recommendation attempt for this user (if any): $\langle Ref_{i,u} \rangle$. There are now $|C_u|$ candidate items: $\langle C_u \rangle$. Please consider the user's historical interaction sequence (and reflections), select appropriate items from the candidates, and rank them to recommend to the user. Think step by step. Your recommendations:

3.2 Overall Framework

We propose MoRE, a novel reflection-based framework, to model the dynamic preferences in LLM-based SeqRec. As shown in Fig. 3, MoRE consist of two key modules:

Multi-Perspective Reflectors (offline reflection). Motivated by the need to decouple explicit/implicit sequential features from user interaction histories and integrate user-item-user collaborative patterns, MoRE employs three perspective-aware reflection processes:

(1) Explicit/Implicit Decoupling: Two distinct reflections analyze a single user's history to disentangle explicit preferences (e.g., item titles and descriptions) and implicit preferences (e.g., attribute-level sequential patterns).

(2) CF-Aware Integration: A third reflection captures cross-user collaborative signals (e.g., rating trends and user similarity) through item-centric analysis.

As discussed in Sec. 1, these derived reflections based on historical interactions are hints to enhance LLM's future recommendations. The details of this module will be introduced in Sec. 3.3.

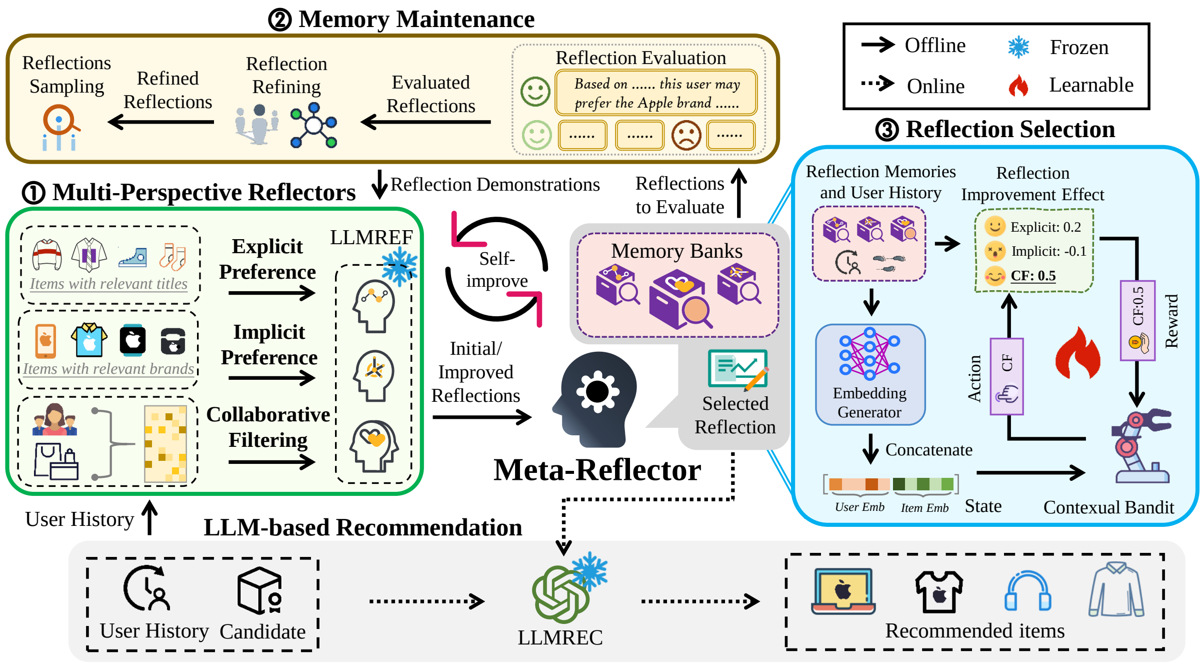


Figure 3: The overall architecture of MoRE framework. MoRE incorporates multi-perspective reflectors to generate reflections offline ① and a meta-reflector to self-improve these reflections ② and learn to select the most appropriate one ③. During online recommendation ④, the meta-reflector selects suitable reflections for the current user.

Meta-Reflector (offline iteration and online selection). This module has two main objectives. First, it maintains three memory banks for each user to collect reflections from the corresponding three reflectors and uses a refining-and-iteration strategy to self-improve the reflections and update the current reflection memories for each user. Second, it is responsible for deciding which of the three perspectives of reflection should be chosen to enhance the current recommendation at each step for the user. This decision-making process is formulated as a contextual bandit problem [16]. These two parts will be elaborated in Sec. 3.4 and Sec. 3.5.

3.3 Offline: Multi-Perspective Reflections

As illustrated on the left of Fig. 3, MoRE aims at exploiting users' historical interaction to generate reflections from the perspectives of explicit preference, implicit preference, and collaborative signals. We achieve this goal through the following reflectors:

Explicit Preference (EP) Reflector captures user u's explicit preferences by leveraging the LLM to reflect on the user's interaction history $\mathcal{S}_u$, the candidate items C_u^t, and the discrepancy between the predicted ranking $\hat{O}_u^t$ generated from LLM-based recommender and the ground-truth v_u^t. The reflection $Ref_{\mathrm{EP},u}$ from the perspective of explicit preference can be formulated as follows:

$$Ref_{EP,u} = \mathrm{LLM}_{\mathrm{REF}}(P_{EP}(u, \mathcal{S}_u^{t-1}, C_u^t, \hat{O}_u^t, v_u^t)), \tag{6}$$

where P_{EP} denotes the prompt specifically designed to guide the LLM in reflecting explicit user preferences, defined as follows:

EP reflection prompt P_{EP}

*You are a reflector for an LLM-based recommender system, understanding the explicit preferences embodied in the user's historical sequence and analyzing the areas for improvement in past recommendation attempts to provide reflections for the future. Explicit preferences are derived from an analysis of **recent tendencies reflected in the sequence of item titles and descriptions** within the user's history.*
You need to:
***1.** Analyze the history with associated text to identify explicit preferences.*
***2.** Analyze the logic and rationale behind past recommendation attempts.*
***3.** Examine potential shortcomings in the past and provide suggestions for improvement.*
User's historical sequence with related description: $\langle REP_{EP}(\mathcal{S}_u^{t-1})\rangle$. Candidates: $\langle REP_{EP}(C_u^t)\rangle$. Past recommendation attempts: $\langle REP_{EP}(\hat{O}_u^t)\rangle$. User new interaction (if any): $\langle REP_{EP}(o_u^t)\rangle$. Historical reflection demonstrations (if any): $\langle DEMO\rangle$. Your reflection:

Please note that in EP reflector, item titles with descriptions are used to represent $\mathcal{S}_u^{t-1}, C_u^t, \hat{O}_u^t$, and v_u^t through $REP_{EP}()$.

Implicit Preference (IP) Reflector is designed to capture the preference embedded in the other item attributes (e.g., item brands). Similar to the EP reflector, the IP reflector takes the interaction history $\mathcal{S}_u$ of user u, the candidate item list C_u, the predicted ranking list $\hat{O}_u$, and the ground-truth item v_u^t as input, and produce the reflection through a frozen LLM:

$$Ref_{IP,u} = \text{LLM}_{\text{REF}}(P_{IP}(u, \mathcal{S}_u^{t-1}, C_u^t, \hat{O}_u^t, v_u^t)), \tag{7}$$

where P_{IP} denotes the prompt for guiding the LLM in reflecting implicit user preferences, defined as follows:

IP reflection prompt P_{IP}

*You are a reflector for an LLM-based recommender system, understand the **implicit preferences** embodied in the user's historical sequence and analyze the areas for improvement in past recommendation attempts to provide reflections for future. Implicit preferences are reflected through the associations shown in subsequences of item attributes within the user's history, such as **subsequences of brands, styles, functions, features, etc.** Requirements:*
***1.** Analyze the logic and rationale behind past recommendation attempts.*
***2.** Focus on attribute subsequences within history to analyze potential associations and causality.*
***3.** Examine possible shortcomings in the past and provide suggestions.*
Historical attribute seq: $\langle REP_{IP}(S_u^{t-1})\rangle$. Candidates: $\langle REP_{IP}(C_u^t)\rangle$. Past recommendation attempts: $\langle REP_{IP}(\hat{O}_u^t)\rangle$. User new interaction (if any): $\langle REP_{IP}(o_u^t)\rangle$. Historical reflection demonstrations (if any): $\langle DEMO\rangle$. Your reflections:

In IP reflector, the attributes of items are used to represent $\mathcal{S}_u^{t-1}, C_u^t, \hat{O}_u^t$, and v_u^t through $REP_{IP}()$.

Collaborative Filtering (CF) Reflector aims to integrate patterns and similarities across user interaction sequences into LLM-based recommender systems. Unlike approaches that introduce new token embeddings into the vocabulary [19, 44] or insert special sequences into prompts [41], we leverage a pre-trained collaborative filtering model $\mathcal{M}_u$ to incorporate CF ratings into the LLM reflection process. Specifically, the CF reflector takes as input the target user u's interaction history $\mathcal{S}_u^{t-1}$, the candidate item set C_u^t, the predicted ranking $\hat{O}_u^t$, and the ground-truth item v_u^t, and generates reflections using a frozen LLM:

$$Ref_{CF,u} = \text{LLM}_{\text{REF}}(P_{CF}(u, \mathcal{S}_u^{t-1}, C_u^t, \hat{O}_u^t, v_u^t)), \tag{8}$$

where P_{CF} is the prompt designed to guide the LLM in reflecting CF signals. The prompt structure is detailed as follows:

CF reflection prompt P_{CF}

*You are a reflector for an LLM-based recommender system, utilizing a Collaborative Filtering (CF) model to obtain items' **CF signals** and analyzing potential improvement in past recommendation attempts for future suggestions. The sequence of historical items' CF ratings reflects **patterns and similarities across user sequences**. Your step-by-step task:*
***1.** Analyze historical CF ratings to identify preferences, trends, or interest shifts.*
***2.** Evaluate past recommendation effectiveness by comparing suggested items with actual user interactions.*
***3.** Analyze potential shortcomings in past recommendation attempts and provide suggestions for improvement.*
Items presented with CF ratings: $\langle REP_{\mathcal{M}_u}(S_u^{t-1}, C_u^t, \hat{O}_u^t, o_u^t)\rangle$ Effective historical reflection demonstrations (if any): $\langle DEMO\rangle$. Your reflection:

In P_{CF}, $\mathcal{S}_u^{t-1}, C_u^t, \hat{O}_u^t$, and v_u^t are represented by their corresponding CF ratings generated by $\mathcal{M}_u$ which is denoted as $<REP_{\mathcal{M}_u}>$.

For each user, we maintain three distinct memory banks, each dedicated to storing the generated reflections from one of the three perspectives (Explicit Preference, Implicit Preference, and Collaborative Filtering respectively). Each memory bank for a user stores a collection of past reflections for that user from the specific perspective. These memory banks are guided by the meta-reflector to improve the reflections, as elaborated below.

3.4 Offline: Reflection Memory Maintenance

Considering that LLMs may not always accurately generate reflection results, we need a mechanism to ensure the validity of reflections after the meta-reflector obtains them from the three reflectors. Therefore, we define a measure called the reflection improvement effect to assess the reflections and devise a refining-and-iteration strategy to update the reflections generated by the LLM.

Reflection Improvement Effect is defined as the difference in LLM recommendation performance on the validation set, with or without the reflections. In other words, if incorporating reflections improves the LLM's performance, it indicates that the reflection is effective for recommendation. The greater the performance improvement, the more effective the reflection. If the reflection improvement effect of a reflection exceeds a preset threshold h, we regard it as an effective one. Formally, we formulate the reflection improvement effect as:

$$\begin{aligned} Imp &= \text{Metric}(\hat{O}_u^{\text{REF}}) - \text{Metric}(\hat{O}_u^{\text{REC}}), \\ \hat{O}_u^{\text{REF}} &= \text{LLM}_{\text{REC}}(P_{\text{REC}}(u, \mathcal{S}_u^{t-1}, C_u^t, Ref_{i,u})), \\ \hat{O}_u^{\text{REC}} &= \text{LLM}_{\text{REC}}(P_{\text{REC}}(u, \mathcal{S}_u^{t-1}, C_u^t)), \end{aligned} \tag{9}$$

where $Ref_{i,u} \in \{Ref_{\text{EP},u}, Ref_{\text{IP},u}, Ref_{\text{CF},u}\}$, Metric denotes the recommendation metric, e.g., NDCG@10.

Refining. As we filter out reflections that offer less or negative SeqRec improvement using the above measure, we further refine the selected reflections with three strategies:

(1) *Global Level:* We select the reflections that bring the greatest average improvement for all users. In this approach, all users share the same reflections.

(2) *Group Level*: We use a CF model to construct embeddings for all users and then cluster them using the K-means++ algorithm [1]. We then choose the reflections that provide the greatest average improvement for users within the same cluster, ensuring that users in the same cluster share the same reflections.

(3) *Individual Level:* We select the reflections that bring the greatest improvement for each user. This strategy leads to personalized reflections.

All of the three strategies are greedy, which may lead to local optima. To mitigate this issue, we propose using the improvement effect brought about by each reflection as the sampling probability. Reflections sampled from the filtered set serve as in-context learning demonstrations. These demonstrations are then used to task the three reflectors with generating improved reflections in subsequent iteration steps, enabling the LLM to self-improve its reflection quality.

Table 1: Statistics of the three pre-processed datasets.

Dataset	#Users	#Items	Avg. len	Sparsity
Arts	55970	22612	8.80	99.96%
Games	55145	17287	9.01	99.95%
Instruments	27404	10450	8.41	99.92%

Iteration. Since both reflection generation and refining are based on users' historical interactions, a major advantage of MoRE is that these two processes can be completed offline. As such, we can further use the sampled reflections as demonstrations and task the three reflectors with generating improved reflections in an in-context learning fashion [5]. In this reflection generation and iteration loop, we enable the LLM to self-improve the reflections. We will further validate the impact of reflection iteration on LLM recommendation performance in the experiments.

3.5 Online: Reflection Perspective Selection

The second objective of the meta-reflector is to model preference shifts in SeqRec. Given three memory banks for each user to collect effective reflections from three different perspectives, we require the meta-reflector to select one of these reflections for recommendation to achieve this goal. We formulate the decision-making process as a Multi-Armed Contextual Bandit (MACB) problem [16], and adopt the Proximal Policy Optimization (PPO) algorithm [22] to efficiently learn to select the reflections. Formally, we define MACB as a tuple $(\mathcal{Z}, \mathcal{A}, \mathcal{R})$, where $\mathcal{Z}, \mathcal{A}, \mathcal{R}$ denote the state space, action (arm) space, and reward function, respectively.

(1) **state** $z \in \mathcal{Z}$ represents recommendation context, constructed by concatenating CF-derived embeddings: $z = [emb_u, emb_v]$ where emb_u is the user embedding and emb_v the average interaction embedding from $\mathcal{S}_u^{t-1}$.

(2) **Action** $\boldsymbol{a} \in \mathcal{A}$ is defined as the selection of a reflection from three memory banks. We represent $\boldsymbol{a}$ as a 3-dimensional vector, with each dimension corresponding to an arm, each arm representing a reflection memory.

(3) **Reward** is defined to assess the recommendation improvement brought by the reflections. We reuse the measure in Eq.(9) and represent the reward as $R(z, \boldsymbol{a}) = Imp$. Since the reflection has been assessed in Sec. 3.3 we do not need to call the LLM here. Additionally, due to the sampling discussed in Sec. 3.4, $R(z, \boldsymbol{a})$ can be both positive and negative, ensuring training efficiency.

(4) **Replay Buffer** is designed to facilitate efficiency in policy optimization. We represent it as $D = (z, \boldsymbol{a}, R(z, \boldsymbol{a}), z')$ to store the tuples of observed state, action, reward, and next state. With the records in the replay buffer, we can refine successful policies and learn from erroneous trials.

Training. We apply the PPO algorithm [22] with an Actor & Critic network and a Clip objective function to efficiently learn MACB.

(1) **Actor & Critic Definition.** To model the selection of reflections and evaluate the value of the states, we implement a policy network parameterized by MLPs to define our selector's Actor $\pi_{\boldsymbol{\theta}}$ and Critic $V_{\boldsymbol{\psi}}$, which maps the environmental space $\mathcal{Z}$ to the action space $\mathcal{A}$ and reward function R respectively:

$$\begin{aligned} &\text{Actor } \pi_{\boldsymbol{\theta}}(z) : \boldsymbol{a} = \text{softmax}(\boldsymbol{\theta} \cdot z), \\ &\text{Critic } V_{\boldsymbol{\psi}}(z) : r = \boldsymbol{\psi} \cdot z, \end{aligned} \tag{10}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ are the parameters of Actor and Critic respectively.

(2) **Clip Objective.** During the training process, the selector's Actor uses the ϵ-greedy ($\epsilon = 0.1$) strategy to explore the environment, with probability ϵ to take a random action while with probability $(1 - \epsilon)$ to exploit the learned policy $\pi_{\boldsymbol{\theta}}$. We adopt the clip objective function for PPO training to maximize the reward selection as follows:

$$\mathcal{L} = \min\left(\frac{\pi_{\boldsymbol{\theta}}}{\pi_{\text{old}}} A(z, \boldsymbol{a}), clip(\frac{\pi_{\boldsymbol{\theta}}}{\pi_{\text{old}}}, 1 - \delta, 1 + \delta) A(z, \boldsymbol{a})\right), \tag{11}$$

where $A(z, \boldsymbol{a}) = \mathbb{E}(R(z, \boldsymbol{a})) - V_{\boldsymbol{\psi}}(z)$ is advantage function used to evaluate z's cumulative rewards $\mathbb{E}(R(z, \boldsymbol{a}))$ and current value $V_{\boldsymbol{\psi}}(z)$. δ is clip threshold.

Inference. During inference, we select the reflection memories with the highest confidence from the Actor, exploit the reflection $Ref_{i,u}$ with the highest Imp score in the memory, and make a recommendation according to Eq.(2) using the prompt P_{REC}.

4 Experiment

In this section, we conduct extensive experiments to answer the following research questions:

- **RQ1:** How does MoRE compare to various existing baselines in terms of recommendation performance?
- **RQ2:** How effective are the multi-perspective reflections and reflection perspective selection in MoRE?
- **RQ3:** For reflection memory maintenance, how effective are the refining and iteration strategies?
- **RQ4:** Does MoRE have an advantage in terms of training cost?

4.1 Experiment Settings

Dataset. Following [44], we conduct our experiments on three subsets of the Amazon dataset [21]: "Arts", "Video", and "Instruments"[1]. Each item in these subsets contains attributes that capture both explicit user preferences (e.g., item titles) and implicit preferences (e.g., attributes like brands and functional traits). We follow [44] to preprocess the data to achieve a fair comparison, with the statistical information shown in Tab. 1.

Baselines. We adopt three types of SeqRec methods as baselines: (1) Traditional Deep Learning Models: Caser [30] captures local and global patterns via CNN; GRU4Rec [10] models sequences with GRUs; SASRec [13] and BERT4Rec [28] employ uni-/bidirectional Transformers for next-item prediction; FDSA [40] models item-feature transitions via dual attention. (2) Training-Free LLM Methods: LLM4RS [4] aligns LLMs with IR ranking strategies; LLM-Rank [12] achieves zero-shot ranking via specialized prompting. (3) Training-Required LLM Methods: LC-Rec [44] expands vocabulary for collaborative filtering; TALLRec [2] uses instruction tuning with text descriptions; BinLLM [41] encodes user-item interactions as binary sequences; Re2LLM [31] employs fixed reflection pools for

[1]Note that "Arts", "Games" and "Instruments" represent the abbreviations of "Arts, Crafts and Sewing", "Video Games" and "Musical Instruments" respectively in the Amazon Review dataset (2018).

Table 2: Recommendation performance comparison. All LLM-based methods utilize Llama-3 and ensure an identical candidate set for a fair comparison. The best and the second-best performances are denoted in bold and underlined fonts, respectively. "N@K" is short for "NDCG@K". The "Imp." indicates the percentage improvement of MoRE over the best performances from baselines. "-" means that the baseline can only output a top-1 prediction and cannot rank or recall multiple items. † denotes MoRE performs significantly better than baselines based on two-tailed paired t-test with Bonferroni correction ($p < 0.05$).

Methods	Arts					Games					Instruments				
	HR@1	HR@5	HR@10	N@5	N@10	HR@1	HR@5	HR@10	N@5	N@10	HR@1	HR@5	HR@10	N@5	N@10
Caser	0.103	0.2555	0.3603	0.1823	0.2161	0.0899	0.3024	0.4397	0.1997	0.2440	0.1349	0.2931	0.3968	0.2193	0.2527
FDSA	0.1211	0.2536	0.3670	0.1875	0.2241	0.1196	0.3073	0.4466	0.2150	0.2597	0.1403	0.2842	0.3905	0.2157	0.2497
BERT4Rec	0.0896	0.2402	0.3613	0.1680	0.2072	0.0899	0.2796	0.4160	0.1861	0.2299	0.1233	0.2618	0.3789	0.1954	0.2326
GRU4Rec	0.1001	0.2431	0.3708	0.1722	0.2129	0.0929	0.3192	0.4644	0.2083	0.2554	0.1340	0.3021	0.4021	0.2201	0.2520
SASRec	0.1850	0.3570	0.4280	0.2788	0.3016	0.1900	0.2915	0.4348	0.1898	0.2369	0.1832	0.3327	0.3701	0.2594	0.2809
LLM4RS	0.0543	0.1087	0.1277	0.0826	0.0887	0.0662	0.1976	0.2233	0.1338	0.1421	0.0330	0.0900	0.1188	0.0624	0.0717
LLMRank(CoT)	0.1725	0.3337	0.4061	0.2595	0.2828	0.1611	0.3508	0.4496	0.2607	0.2925	0.1457	0.2904	0.3709	0.2202	0.2461
LC-Rec	0.2145	0.3768	0.4309	0.3006	0.3181	**0.2599**	0.4140	0.4646	0.3426	0.3593	0.1592	0.3296	0.3920	0.2486	0.2691
Re2LLM	0.2736	0.4252	0.4938	0.3535	0.3757	0.2223	0.4180	0.5198	0.3231	0.3559	0.2011	0.3521	0.4424	0.2787	0.3074
MoRE	**0.2898**	**0.4376**	**0.5043**	**0.3705**	**0.3922**	0.2569	**0.4773**	**0.5731**	**0.3647**	**0.3957**	**0.2163**	**0.3824**	**0.4674**	**0.2981**	**0.3251**
Imp.	5.92%†	2.92%†	2.13%†	4.81%†	4.39%†	-1.15%	14.19%†	10.25%†	6.45%†	10.13%†	7.56%†	8.61%†	5.65%†	6.96%†	5.76%†

suggestions; LLaRA [19] combines user knowledge and behavior patterns via LoRA; A-LLMREC [14] aligns CF embeddings with LLM vocabulary.

Evaluation Metrics. We adopt two widely used metrics, top-k Hit Ratio (HR@k) and Normalized Discounted Cumulative Gain (NDCG@k), with $k \in \{5, 10\}$. We follow [44, 45] to employ the leave-one-out strategy for the obtaining of training, validation, and test data. Specifically, for each user behavior sequence, the last item is used as the test data, the penultimate item is used as the validation data, and the remaining interaction records are used for training. Following [12, 14, 19, 31], we randomly sample $|C_u| - 1$ negative items to construct $|C_u|$ candidates with 1 target item for each user. Following [31], $|C_u|$ is set to 50 for all methods. All methods except those **inherently** limited to top-1 prediction (without ranking) must provide top-10 ranked items.

Implementation Details. We use LLaMa-3-8B-Instruct [6] as the backbone for all LLM-based methods. Following [2, 12, 14, 31], we respectively perform random sampling from three datasets for efficiency, and exclude users with overly long interaction sequences and then randomly we sample 1,000 users, with items remaining unchanged. For simplicity, we trained DMF[2] [32] with an embedding size of 64 to provide CF scores for reflection (Sec. 3.3) and to assist with user clustering within refining (Sec. 3.4). The number of clusters is set to 20. Threshold h is set to 0.1. We use RecBole [43] to implement all traditional SeqRec baselines (e.g. SASRec) with the Adam optimizer and grid search[3]. For LLM baselines, we follow the origin settings and make proper adaptions to our scenario. For constrained generative retrieval baselines, we construct the constraints following the corresponding setting. For baselines that only provide top-1 prediction, we provide the same candidate set as MoRE and, following the original settings, do not require predictions beyond the top-1, thereby obtaining their best top-1 results under a lower task difficulty. For baselines limited to yes/no outputs, we adjust them, following LLaRA [19], to predict the next item. All experiments are run on a 4×A800-PCIE-80GB GPUs server.

[2]Please note that other CF models can be easily incorporated into MoRE.

[3]Grid search space was set as: *embedding size* : {32, 64, 128}, *learning rate* : $\{1e^{-2}, 1e^{-3}, 1e^{-4}\}$ and *batch size* : {1024, 2048, 4096}.

Table 3: Comparison of HR@1 Performance with LLM baselines that inherently only provide top-1 prediction.

Methods	Arts	Games	Instruments
TALLRec	0.2559	0.1975	0.1588
BinLLM	0.2401	0.2347	0.1875
LLaRA	0.2803	0.2263	0.1763
A-LLMREC	0.2193	0.2342	0.1897
MoRE	**0.2898**	**0.2569**	**0.2163**

4.2 Recommendation Performance Comparison

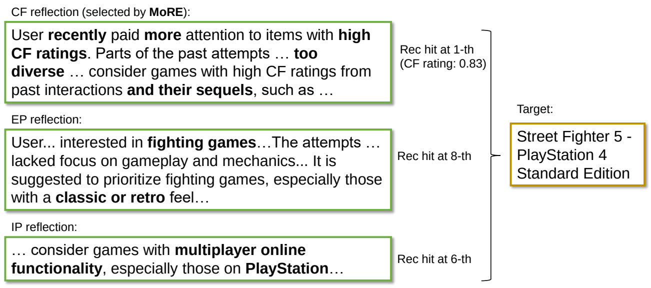
To answer **RQ1**, we compare the SeqRec performance of MoRE against baselines for 10 trials, with the average results presented in Tab. 2 and Tab. 3. We have the following observations: (1) MoRE outperforms all baselines across all metrics on all datasets, except for the suboptimal performance on the HR@1 on "Games". (2) In general, LLM-based methods demonstrate performance comparable to the traditional methods, which validates the potential of LLMs in SeqRec. (3) Notably, MoRE outperforms LLM-based methods that specialize in predicting one single item (HR@1 only), even when those methods sacrifice ranking capabilities[4].

In summary, MoRE exhibits superior performance in SeqRec, validating the efficacy of our approach. We attribute this to our proposed LLM-based reflection framework, which leverages LLM's analytical and reasoning capabilities to capture users' explicit preferences and implicit behavioral patterns while seamlessly integrating CF knowledge in a clear and comprehensible manner, ultimately enhancing performance in sequence recommendation tasks.

[4]Note that these methods are **inherently** limited to top-1 predictions. Therefore, for these methods, we retain only top-1 results for a fair comparison here.

Table 4: Ablation studies on Amazon Arts. "Random" indicates a random reflection selection from the three memory banks, while "Greedy" denotes that reflection selection is based solely on downstream SeqRec on the validation set. Ref_{all} denotes $Ref_{\text{EP}} + Ref_{\text{IP}} + Ref_{\text{CF}}$. Note: "+" denotes concatenation, which may cause conflicting reflections leading to suboptimal performance. MoRE achieves the best results through user-personalized RL selection.

Methods	HR@1	HR@5	HR@10	N@5	N@10
MoRE	**0.2898**	**0.4376**	**0.5043**	**0.3705**	**0.3922**
- Random	0.2736	0.4023	0.4881	0.3421	0.3701
- Greedy	0.2812	0.4156	0.4957	0.3528	0.3786
- Ref_{EP} only	0.2850	0.4356	0.5033	0.3670	0.3881
- Ref_{IP} only	0.2850	0.4214	0.4881	0.3571	0.3788
- Ref_{CF} only	0.2850	0.4290	0.4871	0.3572	0.3788
- $Ref_{\text{EP}} + Ref_{\text{IP}}$	0.2583	0.4071	0.4700	0.3443	0.3651
- $Ref_{\text{EP}} + Ref_{\text{CF}}$	0.2707	0.3927	0.4623	0.3311	0.3533
- $Ref_{\text{IP}} + Ref_{\text{CF}}$	0.2402	0.3584	0.4156	0.3033	0.3217
- Ref_{all}	0.2393	0.3594	0.4271	0.3051	0.3268


CF reflection (selected by **MoRE**):

User **recently** paid **more** attention to items with **high CF ratings**. Parts of the past attempts … **too diverse** … consider games with high CF ratings from past interactions **and their sequels**, such as …

EP reflection:

User... interested in **fighting games**…The attempts … lacked focus on gameplay and mechanics... It is suggested to prioritize fighting games, especially those with a **classic or retro** feel…

IP reflection:

… consider games with **multiplayer online functionality**, especially those on **PlayStation**…

Figure 4: Example of how LLM reflections from explicit preference (EP), implicit preference (IP), and collaborative filtering (CF) perspectives improve recommendation.

4.3 Ablation Study: Multi-Perspective Reflection and Selection

To validate the effectiveness of each perspective of reflection and answer **RQ2**, we conduct an ablation study on Amazon Arts. The results are presented in Tab. 4.

We find that MoRE, equipped with the reflection perspective selection, outperforms alternatives, demonstrating the effectiveness of the multi-perspective reflections and the corresponding perspective selection. Specifically, MoRE outperforms the variants with not only the single reflection perspective but also the dual or triple perspectives simultaneously. The simultaneous use of multiple perspectives leads to a decrease in performance. This implies potential discrepancies among perspectives and no single perspective of reflection is universally optimal for all users, thereby confirming the necessity for selective choice among perspectives.

Moreover, MoRE's selection strategy surpasses both the Random and Greedy selection approaches which respectively denote a random reflection selection from the three memory banks and the selection solely based on downstream recommendation performance. This highlights the effectiveness of our contextual bandit modeling, enabling MoRE to choose the most suitable reflection for each user.

4.4 Ablation Study: Refining and Iteration

To answer **RQ3**, we test the impact of three refining strategies on SeqRec under different iteration rounds, taking reflections on explicit features as an example. From Fig. 5, we find:

(1) **Iteration**: Overall, as the number of iterations increases, the recommendation performance gradually improves. This indicates that iteration can enhance the quality of reflections, thereby improving the recommendation.

(2) **Refining**: Refining at group and individual levels shows a significantly better trend over that at the global level during iterations, particularly on Arts (Fig. 5(a)). We assume that the global-level reflections may introduce noise from other users, whereas group- and individual-level demonstrations more effectively elicit personalized reflective capabilities from the reflectors.

4.5 Analysis of Computational Cost

To address **RQ4**, we compare training costs (time and GPU memory) across LLM-based baselines using five repeated experiments, reported as Avg. ± SD[5] in Tab. 5. We find: 1) Reflection-based methods (e.g., MoRE) incur lower costs than fine-tuning approaches by avoiding LLM parameter updates; 2) MoRE outperforms Re2LLM in efficiency due to its smaller action space. MoRE's Meta-Reflector selects from 3 reflection perspectives, while Re2LLM trains a PPO-based retrieval agent over a fixed base[6], requiring larger action-space exploration. In summary, MoRE achieves the lowest training costs (time and memory) among all baselines, demonstrating superior training efficiency.

4.6 Case Study

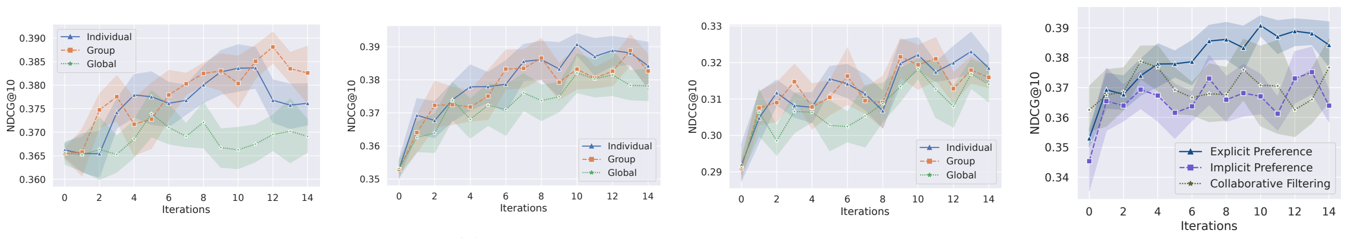
We present a case study demonstrating reflections from EP, IP, and CF perspectives to enhance LLMREC (Fig. 4). The three perspectives improve performance as follows: **EP Reflection** focuses on recent interaction trends (e.g., game genre preferences) to align with explicit preferences. This results in the target item being recommended at the 8-th position (e.g., *Street Fighter 5*), effectively capturing short-term user intentions. **IP Reflection** identifies patterns in historical attributes (e.g., multiplayer functionality, "PlayStation" brand) to adapt to implicit preferences. This moves the target to the 6-th position by analyzing deeper user behavioral patterns. **CF Reflection** leverages collaborative filtering signals and emphasizes highly-rated items, reducing diversity while incorporating historical ratings and relevant sequels. This achieves the top-1 recommendation position. The **Meta-reflector** dynamically selects the optimal reflection (CF in this case), enabling MoRE to deliver optimal recommendations.

The reflection mechanism bridges the semantic gap between LLMs and CF: LLMs utilize simple score comparisons instead of interpreting embeddings. This approach cost-effectively equips LLMs with domain-specific knowledge (e.g., CF signals) while mitigating their adaptation challenges in specialized domains.

[5]Average and standard deviation.
[6]The size is 20.

Table 5: Training cost comparisons among LLM-based methods on four A800. "Training time" is measured in hours, shown as the mean and standard deviation (SD). "VRAM" denotes the Peak VRAM Usage, measured in GB. MoRE's RL framework minimizes computation cost through compact action space ($|\mathcal{A}| = 3$, 3 perspectives).

Methods	Arts		Games		Instruments	
	Training time	VRAM	Training time	VRAM	Training time	VRAM
TALLRec (Fine-Tune)	26.228(±0.818)	46.98	18.018(±0.290)	43.24	18.338(±0.487)	42.95
BinLLM (Fine-Tune)	10.209(±0.540)	124.67	11.468(±0.328)	120.57	10.435(±0.451)	115.48
LLaRA (Fine-Tune)	9.013(±0.161)	77.92	6.405(±0.281)	65.83	6.863(±0.187)	74.36
A-LLMREC (Fine-Tune)	56.975(±1.621)	217.74	55.975(±1.058)	202.09	56.316(±1.992)	231.63
LC-Rec (Fine-Tune)	20.382(±0.670)	206.37	21.361(±0.678)	201.35	13.720(±0.436)	199.93
Re2LLM (RL, $\|\mathcal{A}\| = 20$)	4.007(±0.007)	2.77	3.894(±0.005)	1.99	4.152(±0.008)	1.13
MoRE (RL, $\|\mathcal{A}\| = 3$)	**3.982(±0.001)**	**2.77**	**3.854(±0.002)**	**1.99**	**4.093(±0.003)**	**1.13**



(a) Arts **(b) Games** **(c) Instruments** **(d) Iteration for 3 Perspectives**

Figure 5: Impact of refining and iteration strategies on recommendation performance. We repeated the experiment 5 times, with the mean value for each method represented by the line and the shaded area indicating the standard deviation. (a), (b), and (c) demonstrate the effects of iterations and different levels of refining on three datasets, while (d) shows the effects of the three perspectives with Games as an example.

5 Conclusion

In this paper, we propose a mixture of reflectors framework, namely MoRE, for modeling and learning the dynamic user preferences in SeqRec. We first introduce three reflectors for each user to generate LLMs' reflections from the perspectives of explicit user preferences, implicit user preferences, and collaborative signals. Building on these reflectors, we introduce a meta-reflector that evaluates and updates the generated reflections using a self-improving strategy. It selects the most appropriate perspective and corresponding reflections for each user's current recommendation using a contextual bandit algorithm. Furthermore, MORE's unique self-improving meta-reflector, through its rigorous refining and iteration strategies, explicitly addresses the critical challenge of ensuring the validity and quality of LLM-generated reflections, providing reliable hints for recommendation. This mechanism effectively bridges the semantic understanding of LLMs with established recommendation principles, enhancing both performance and interpretability. Extensive experiments conducted on three benchmarks demonstrate that MoRE consistently outperforms traditional recommendation methods and LLM-based recommendation methods with minimal GPU memory usage and training time overhead.

Acknowledgments

This work was funded by the National Key R&D Program of China (2023YFA1008704), the National Natural Science Foundation of China (62472426,62376275), the Beijing Social Science Foundation Planning Project (Grant No. 24GLC041), the Fundamental Research Funds for the Central Universities in UIBE (Grant No. 24QN06, 24PYTS22). Supported by fund for building world-class universities (disciplines) of Renmin University of China. Work partially done at Beijing Key Laboratory of Research on Large Models and Intelligent Governance, and Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE. The work was done while Weicong Qin, Yi Xu and Chenglei Shen were interns at Lenovo.

References

[1] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. 2012. Scalable k-means++. *Proc. VLDB Endow.* 5, 7 (mar 2012), 622–633. https://doi.org/10.14778/2180912.2180915

[2] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. *arXiv preprint arXiv:2305.00447* (2023).

[3] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084* (2022).

[4] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1126–1132.

[5] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234* (2022).

[6] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783* (2024).

[7] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*. 299–315.

[8] Zhankui He, Handong Zhao, Zhe Lin, Zhaowen Wang, Ajinkya Kale, and Julian McAuley. 2021. Locker: Locally constrained self-attentive sequential recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3088–3092.

[9] Zhankui He, Handong Zhao, Zhaowen Wang, Zhe Lin, Ajinkya Kale, and Julian Mcauley. 2022. Query-Aware Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4019–4023.

[10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[11] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. Large language models are zero-shot rankers for recommender systems. *arXiv preprint arXiv:2305.08845* (2023).

[12] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*. Springer, 364–381.

[13] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*.

[14] Sein Kim, Hongseok Kang, Seungyoon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. 2024. Large language models meet collaborative filtering: An efficient all-round llm-based recommender system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1395–1406.

[15] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *CIKM*.

[16] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*. 661–670.

[17] Muyang Li, Zijian Zhang, Xiangyu Zhao, Wanyu Wang, Minghao Zhao, Runze Wu, and Ruocheng Guo. 2023. AutoMLP: Automated MLP for Sequential Recommendations. In *Proceedings of the ACM Web Conference 2023*. 1190–1198.

[18] Muyang Li, Xiangyu Zhao, Chuan Lyu, Minghao Zhao, Runze Wu, and Ruocheng Guo. 2022. MLP4Rec: A Pure MLP Architecture for Sequential Recommendations. In *31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence (IJCAI-ECAI 2022)*. International Joint Conferences on Artificial Intelligence, 2138–2144.

[19] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-recommendation assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1785–1795.

[20] Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2023. A Multi-facet Paradigm to Bridge Large Language Model and Recommendation. *arXiv preprint arXiv:2310.06491* (2023).

[21] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP-IJCNLP*.

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

[23] Chenglei Shen, Xiao Zhang, Wei Wei, and Jun Xu. 2023. Hyperbandit: Contextual bandit with hypernewtork for time-varying user preferences in streaming recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2239–2248.

[24] Chenglei Shen, Jiahao Zhao, Xiao Zhang, Weijie Yu, Ming He, and Jianping Fan. 2024. Generating Model Parameters for Controlling: Parameter Diffusion for Controllable Multi-Task Recommendation. *arXiv preprint arXiv:2410.10639* (2024).

[25] Teng Shi, Zihua Si, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Dewei Leng, Yanan Niu, and Yang Song. 2024. UniSAR: Modeling User Transition Behaviors between Search and Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1029–1039.

[26] Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Han Li. 2025. Retrieval Augmented Generation with Collaborative Filtering for Personalized Text Generation. *arXiv preprint arXiv:2504.05731* (2025).

[27] Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Enyun Yu. 2025. Unified Generative Search and Recommendation. *arXiv preprint arXiv:2504.05730* (2025).

[28] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*.

[29] Jiakai Tang, Sunhao Dai, Teng Shi, Jun Xu, Xu Chen, Wen Chen, Wu Jian, and Yuning Jiang. 2025. Think before recommend: Unleashing the latent reasoning power for sequential recommendation. *arXiv preprint arXiv:2503.22675* (2025).

[30] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*.

[31] Ziyan Wang, Yingpeng Du, Zhu Sun, Haoyan Chua, Kaidong Feng, Wenya Wang, and Jie Zhang. 2024. Re2LLM: Reflective Reinforcement Large Language Model for Session-based Recommendation. *arXiv preprint arXiv:2403.16427* (2024).

[32] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep matrix factorization models for recommender systems.. In *IJCAI*, Vol. 17. Melbourne, Australia, 3203–3209.

[33] Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, et al. 2023. Retroformer: Retrospective large language agents with policy gradient optimization. *arXiv preprint arXiv:2308.02151* (2023).

[34] Zhenrui Yue, Sara Rabhi, Gabriel de Souza Pereira Moreira, Dong Wang, and Even Oldridge. 2023. LlamaRec: Two-Stage Recommendation using Large Language Models for Ranking. *arXiv preprint arXiv:2311.02089* (2023).

[35] Changshuo Zhang, Sirui Chen, Xiao Zhang, Sunhao Dai, Weijie Yu, and Jun Xu. 2024. Reinforcing Long-Term Performance in Recommender Systems with User-Oriented Exploration Policy. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1850–1860.

[36] Changshuo Zhang, Teng Shi, Xiao Zhang, Qi Liu, Ruobing Xie, Jun Xu, and Ji-Rong Wen. 2024. Modeling Domain and Feedback Transitions for Cross-Domain Sequential Recommendation. *arXiv preprint arXiv:2408.08209* (2024).

[37] Changshuo Zhang, Xiao Zhang, Teng Shi, Jun Xu, and Ji-Rong Wen. 2025. Test-Time Alignment for Tracking User Interest Shifts in Sequential Recommendation. *arXiv preprint arXiv:2504.01489* (2025).

[38] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. *arXiv preprint arXiv:2305.07001* (2023).

[39] Kepu Zhang, Teng Shi, Sunhao Dai, Xiao Zhang, Yinfeng Li, Jing Lu, Xiaoxue Zang, Yang Song, and Jun Xu. 2024. SAQRec: Aligning Recommender Systems to User Satisfaction via Questionnaire Feedback. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 3165–3175.

[40] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, Xiaofang Zhou, et al. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation.. In *IJCAI*.

[41] Yang Zhang, Keqin Bao, Ming Yan, Wenjie Wang, Fuli Feng, and Xiangnan He. 2024. Text-like Encoding of Collaborative Information in Large Language Models for Recommendation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 9181–9191. https://doi.org/10.18653/v1/2024.acl-long.497

[42] Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023. CoLLM: Integrating Collaborative Embeddings into Large Language Models for Recommendation. *arXiv preprint arXiv:2310.19488* (2023).

[43] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *proceedings of the 30th acm international conference on information & knowledge management*. 4653–4664.

[44] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 1435–1448.

[45] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022*. 2388–2399.

[46] Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. 2024. Collaborative large language model for recommender systems. In *Proceedings of the ACM Web Conference 2024*. 3162–3172.