Tunable LLM-based Proactive Recommendation Agent

Mingze Wang[1], Chongming Gao[1†], Wenjie Wang[1†], Yangyang Li[2], Fuli Feng[1]

[1]University of Science and Technology of China, [2]Academy of Cyber, Beijing, China
gnaweinre@mail.ustc.edu.cn, {chongming.gao, wenjiewang96}@gmail.com,
liyangyang@live.com, fulifeng93@gmail.com

Abstract

Recommender systems are indispensable on various digital platforms. However, traditional methods often reinforce existing user interests, which leads to echo chambers and limits diversity. Proactive Recommendation Systems (PRS) aim to address this issue by cultivating users' latent interests through multi-step recommendations. Despite advancements, challenges persist particularly in optimizing long-term rewards and adapting to real-time user feedback. In this study, we propose an LLM-based Actor-Critic Agent framework to enhance PRS. This framework utilizes the LLM-based agent to adjust recommendations in real time based on feedback and employs agent-tuning methods to optimize long-term rewards using three proposed reward functions. Extensive experiments validate the significant superiority of this framework over existing methods by optimizing long-term rewards and dynamically evolving with user feedback. Our codes are available at `https://github.com/gnaWeinrE/T-PRA`.

1 Introduction

Recommender systems are inherently text-intensive, necessitating substantial natural language processing to interpret item content (An et al., 2019; Bhagavatula et al., 2018) and recognize user interests (Cheng et al., 2023). Current approaches (Bao et al., 2024) predominantly focus on catering to the recognized user interest, often leading to polarized recommendation distributions. Over time, this will exacerbate the filter bubble effect (Areeb et al., 2023; Gao et al., 2023a,b), which continually narrows user interests, undermining the long-term health of recommendation ecosystems. In contrast, users' latent interests are broad and developable as they consume content (Wang et al., 2024). As such, *Proactive Recommendation* (Bi et al., 2024a; Wang et al., 2025b; Lian et al., 2025a) becomes a promising research direction, aiming to cultivate

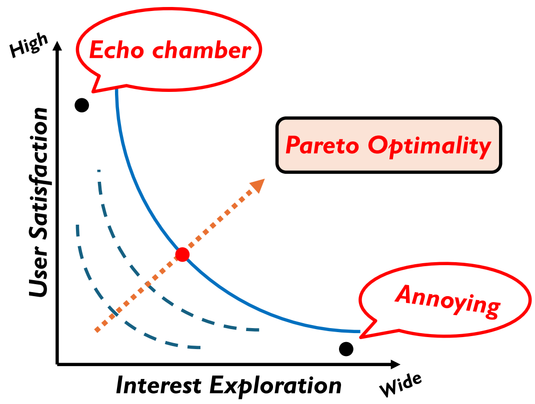

Figure 1: Trade-off between user satisfaction and user interest exploration. Catering to the recognized user interest enhances user satisfaction but causes filter bubble phenomenon. Conversely, randomly displaying advertisements to broaden user interests becomes annoying and negatively impacts user satisfaction.

latent interest throughout the recommendation process without compromising the overall user experience.

Existing research on proactive recommendation mainly cultivates users' latent interest through multi-step recommendation strategies. Current works, such as the Influential Recommender System (IRS) (Zhu et al., 2023), use a Transformer to learn dynamic patterns of user interest and cultivate latent interests by exposing multiple items. Later, Wang et al. (2025a) incorporates the powerful instruction-following and planning capabilities of Large Language Models (LLMs). They propose LLM-IPP, achieving state-of-the-art (SOTA) performance compared to the IRS. However, significant limitations remain: 1) cultivating user interests requires a long-term recommendation strategy while LLM-IPP cannot optimize long-term cumulative rewards. 2) LLM-IPP lacks flexibility, as it cannot adapt to real-time user feedback and has the risk of repeatedly suggesting unappealing content.

To address the limitations, we distill two key ob-

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 19262–19276
July 27 - August 1, 2025

jectives for developing an effective *proactive recommendation system* (PRS). (1) The PRS should be flexible to update the recommendation strategy in real time based on user feedback. (2) The PRS should optimize the long-term overall rewards of its strategy. This involves fostering user interest in some target items that were previously less favored while minimizing any adverse impacts on the user's experience, as illustrated in Figure 1. To enhance user interest, the recommendation strategy must maintain coherence and gradually align with the user's interests and target items. Additionally, reducing the negative impact requires ensuring users' acceptance toward the recommended items.

Given the aforementioned objectives, we conceptualize proactive recommendation as a complex path-planning problem in sequential decision-making. The key to this problem lies in leveraging the inherent capabilities of LLMs, particularly their extensive world knowledge and sophisticated reasoning abilities. To effectively handle real-time user feedback and dynamically adapt recommendation strategies, we design an LLM-based agent system consisting of an *Actor* and an *Advisor*. The Actor is responsible for generating immediate recommendations, while the Advisor operates at a slower, more deliberate pace — collecting users' real-time feedback, reasoning user interest dynamics, and adapting the current strategy to guide the Actor's decision-making process.

While existing LLMs offer strong foundational capabilities, they are not inherently optimized for proactive recommendation tasks, often leading to suboptimal performance, see evidence in Section 5.5.1. To address this limitation, we employ an agent-tuning approach to refine LLMs with the goal of maximizing long-term rewards. Specifically, we propose the **T**unable LLM-based **P**roactive **R**ecommendation **A**gent (T-PRA), which introduces an LLM-based *Critic* to evaluate the quality of recommendations. The Critic assigns an advantage value to each recommendation, reflecting its alignment with the long-term objectives of the proactive recommendation task. These advantage values enable the construction of preferred and dispreferred actions and thoughts for both the Actor and Advisor. Using these comparisons, we apply Direct Preference Optimization (DPO) (Rafailov et al., 2023) to fine-tune the Actor and Advisor, enabling the LLMs to generate more favorable and strategically aligned outputs.

To ensure the Critic produces accurate advantage values that effectively capture the long-term rewards of proactive recommendations, we design three reward functions during the value learning process, capturing key objectives of PRS: *coherence* in recommendation sequences, *distance to the target item*, and *user acceptance*. By integrating these reward functions, T-PRA effectively aligns recommendation strategies with both the goal of flexible strategies and long-term reward optimization. In conclusion, our contributions are threefold:

- We propose an LLM-based proactive recommendation agent capable of dynamically incorporating real-time user feedback while optimizing for long-term rewards in proactive recommendation.
- We introduce a novel agent-tuning approach that enhances the ability of LLMs to generate coherent, target-aligned, and user-accepted recommendations, effectively cultivating user interest while minimizing negative impacts on user experience.
- Lastly, we design and conduct a series of experiments to validate the effectiveness of the proposed framework for proactive recommendation tasks, demonstrating an average increase of 38% in effectiveness to enhance user interest in a particular item compared to the SOTA methods.

2 Related Works

2.1 LLM-based Agent

Benefiting from recent progress in LLMs (Xu et al., 2025), LLM-based agents integrate LLMs with autonomous decision-making processes, enabling them to perform complex tasks across various domains. Certain research focuses on agent planning, leveraging language models to formulate action sequences aligned with specific goals and environmental limitations, utilizing techniques such as In-Context Learning (Kojima et al., 2022; Yao et al., 2023; Zheng et al., 2024) and external planning algorithms(Hao et al., 2023; Romero et al., 2024; Lawrynowicz et al., 2024). Additional investigations concentrate on enhancing agent memory (Singla et al., 2021; Zhao et al., 2024), which plays a pivotal role in refining decision-making processes. This involves the exploration of long-term memory systems like retrieval-augmented generation (Lewis et al., 2020; Zhang et al., 2023; Deng et al., 2024) to improve data retrieval and management. Furthermore, the concept of reflective thinking (Shinn et al., 2023; Wang et al., 2023) allows agents to critically assess their actions and assimilate feedback from the environment, thereby increasing adaptability. Additionally, some research working on multi-agent systems (Hong et al., 2024; Chan et al., 2024) aims to optimize interaction and communication among agents to enable the completion of sophisticated tasks through cooperation or competition. In our study, we integrate robust planning and decision-making functionalities of LLMs to provide timely and appropriate recommendations based on previous feedback.

2.2 Proactive Dialogue

Proactive dialogue systems have made significant improvements in recent years, enabling conversational agents to guide interactions toward specific goals rather than merely responding to user inputs. These systems can be broadly categorized into three types: open-domain, task-oriented, and information-seeking dialogues. In open-domain dialogues, methods like target-guided conversations and prosocial dialogues aim at leading conversations toward designated topics (Konigari et al., 2021; Xie et al., 2021; Mendonça et al., 2024) or maintaining socially responsible interactions by detecting and addressing problematic user behavior (Xu et al., 2021; Kim et al., 2022). In task-oriented dialogues, proactive systems manage different scenarios (Zhou et al., 2020; He et al., 2018), such as negotiations, and enrich task completion by providing useful information (Sun et al., 2021; Chen et al., 2022). In information-seeking dialogues, proactive agents improve the accuracy and relevance of responses by asking clarification questions (Aliannejadi et al., 2019; Deng et al., 2022) or actively eliciting user preferences (Shao et al., 2023; Li et al., 2018) to refine recommendations. These advancements reflect the shift from passive to active engagement in conversational agents, aiming to create more dynamic, efficient, and user-centric interactions. In our research, we utilize proactive agents to analyze user interests and explore these interests by recommending items that users might prefer.

2.3 Proactive Recommendation

Proactive Recommendation aim at exploring users' unobserved latent interests and enhancing users' interest distributions without adversely affecting their overall experience. This concept was first introduced by Zhu et al. (2023), which constructs the proactive recommendation strategy utilizing a Transformer-based sequential model. It learns interest-shifting patterns from users' historical data and encodes the characteristics of target items to generate the strategy. Subsequently, Bi et al. (2024b) is a model-agnostic post-processing strategy that enhances the flexibility of IRS and incorporates an explicit IPG score design, improving the effectiveness of proactive recommendations. However, IPG relies on synthetic rather than real data. Following this, Lian et al. (2025b), a novel multi-round proactive recommendation model, considers the users' intentions to capture broad-level evidence for subsequent steering recommendations, thereby enhancing performance and Wang et al. (2025a) achieves SOTA performance by employing prompt engineering techniques to utilize powerful instruction-following and planning abilities of LLMs for proactive recommendation. In our research, we observed a trade-off between user acceptance and the breadth of user interests. By employing LLM-based agent and agent tuning methods, T-PRA can adjust recommendations based on user feedback and provide recommendations aimed at long-term overall rewards without compromising the overall user experience.

3 Task Definition

Our work builds upon the proactive recommendation task formulation introduced in prior works (Zhu et al., 2023). For a given user, the target item is denoted as i_{Ta}. At step n, the policy π_θ of the PRS generates a sequence of items $\tau_n = \{i_n, i_{n+1}, \dots, i_{Ta}\}$ based on the interaction history $\mathcal{H}_n = \{i_1, i_2, \dots, i_{n-1}\}$, with the goal of gradually expanding the user's interest toward the target item i_{Ta}. Formally, the policy is defined as:

$$\tau = \pi_\theta(\mathcal{H}_n, i_{Ta}). \tag{1}$$

To optimize this task, we adopt a Reinforcement Learning (RL) framework, where an agent learns to make sequential decisions by interacting with an environment. The RL framework is formalized as a Markov Decision Process (MDP) represented by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R})$, adapted to the proactive recommendation setting as follows:

- $\mathcal{S}$ is the state space, encapsulating all possible states the agent may encounter. At step n, the state s_n corresponds to the interaction history: $s_n = \{i_1, i_2, \dots, i_{n-1}\}$.

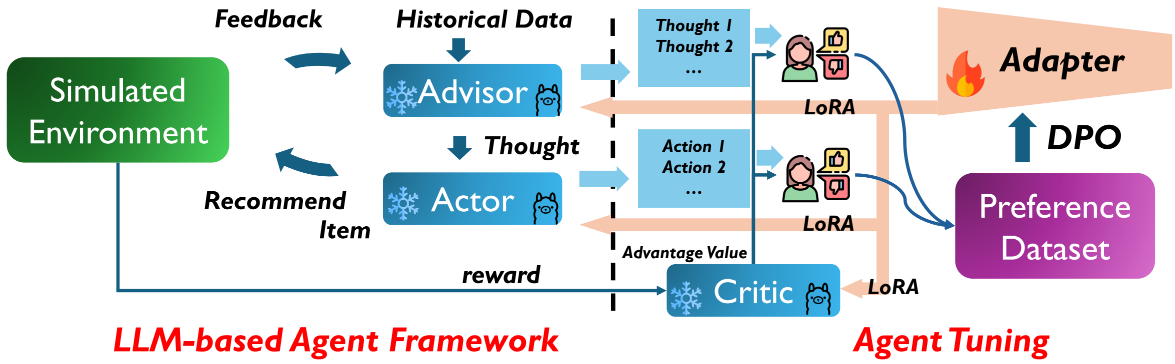

Figure 2: Structure of the tunable LLM-based proactive recommendation agent (T-PRA). The left side represents the LLM-based agent framework, which incrementally generates recommended items and optimizes them flexibly based on feedback from the simulated environment. The right side illustrates the Agent tuning procedure. After collecting a sufficient amount of data, it constructs a preference dataset using a rule-based reward function, with a Critic module, to perform DPO tuning and optimize the long-term rewards.

- $\mathcal{A}$ is the action space, comprising all possible recommendations the agent can make. At step n, the action a_n is the recommended item i_n.
- $\mathcal{R}$ is the reward function, which provides feedback to the agent based on its actions. At step n, the reward r_n reflects the user's real-time feedback to the recommendation i_n.

The recommendation process proceeds iteratively: at each step n, the policy recommends an item i_n, and the interaction history is updated accordingly. This process continues until the target item i_{Ta} is reached, ensuring a gradual and coherent progression of user interest.

4 Methods

In this section, we provide a detailed description of T-PRA. An overview of the framework is illustrated in Figure 2, which consists of two key components:

- **Actor-Advisor Framework**: This component adopts a dual-system approach inspired by the *fast and slow thinking* paradigm (Kahneman, 2011), where the *Actor* makes immediate recommendations, while the *Advisor* performs more strategic reasoning to refine decision-making.
- **Critic-Guided Optimization**: The second component introduces a *Critic* module, which evaluates recommendations and provides structured feedback to optimize the agent's decision-making process. Based on the Critic's signals, DPO is employed to fine-tune both the Actor and the Advisor, enhancing their alignment with long-term proactive recommendation objectives.

Example of Thought

The user's preference score increased after recommending "Call of Juarez", indicating a positive reception. To further guide them toward the target item "Eidolon", I should recommend more open-world action games with a similar gameplay experience.

Figure 3: An example of a thought produced by *Advisor*.

4.1 Actor-Advisor Framework

The Actor-Advisor framework is designed to process information at two levels. Since the Actor relies on signals provided by the Advisor, we first introduce the Advisor module.

4.1.1 Advisor

The Advisor module is responsible for aggregating information from multiple sources, including environmental feedback, user interaction history, and previous recommendations made by the Actor. At step n, the Advisor generates a structured guidance signal, referred to as *thought* t_n, based on the user's historical data $\mathcal{H}_n$, the current state s_n, and the most recent reward from the environment r_{n-1}:

$$t_n = \text{Advisor}(\mathcal{H}_n, s_n, r_{n-1}). \tag{2}$$

We implement the Advisor using an LLM instance, leveraging its ability to synthesize complex information and generate strategic insights. An example of a generated thought is in Fig. 3.

4.1.2 Actor

The Actor is responsible for making direct decisions based on the thought provided by the Advisor. We implement the Actor module using an LLM and adapt its generated recommendations to the dataset using an item grounding process. At each step n, the action a'_n is generated as follows:

$$a'_n = \text{Actor}(s_n, t_n), \tag{3}$$

where s_n is the current state, and t_n is the thought generated by the Advisor.

Item Grounding. Since LLM-generated recommendations may not always align with items in the dataset, we apply an item grounding mechanism to ensure consistency with baseline evaluations. We utilize Llama-3.1-8B-instruct (Dubey et al., 2024) to encode both the generated items and those in the dataset. The final recommendation is selected by identifying the item with the highest cosine similarity to the LLM-generated output:

$$a_n = \arg\max_{a \in I} \ \text{cos\_sim}(e_a, e_{a'_n}), \tag{4}$$

where I represents the dataset items, and e_a denotes the LLM embedding of item a.

4.2 Agent Tuning

To align the Actor and Advisor with the objectives of the proactive recommendation task, we introduce a Critic-guided tuning component. This component evaluates the long-term impact of the Actor's actions and optimizes the Actor and Advisor based on the Critic's judgments.

4.2.1 Critic

The Critic serves as an LLM-based evaluator that estimates the long-term benefits of actions taken by the Actor. Specifically, we employ an LLM instance to compute state values $V(s_n)$, providing a quantitative measure of expected future rewards from a given state:

$$V(s_n) = \text{Critic}(s_n). \tag{5}$$

To mitigate variance in value estimation, we adopt the advantage function (Mnih et al., 2016). The advantage value $A(s_n, a_n)$, which quantifies the relative benefit of selecting action a_n in state s_n, is computed as:

$$A(s_n, a_n) = r_n + \gamma V(s_{n+1}) - V(s_n), \tag{6}$$

where r_n denotes the immediate reward after generating action a_n, and γ is the discount factor that balances short-term and long-term rewards.

4.2.2 Preference Optimization with DPO

With the advantage values indicating the quality of each action, we employ DPO to fine-tune the Actor and Advisor. DPO is an effective preference learning approach that optimizes models to increase the likelihood of preferred outputs while reducing the likelihood of dispreferred ones. To apply DPO, we first construct a preference dataset for actions and thoughts.

At each step n, given the state s_n, we collect all possible actions $\mathcal{A}_n$. Using the advantage values computed for $\mathcal{A}_n$, we identify the most preferred action a_n^w and the least preferred action a_n^l as:

$$\begin{cases} a_n^w = \arg\max_{a \in \mathcal{A}_n} A(s_n, a), \\ a_n^l = \arg\min_{a \in \mathcal{A}_n} A(s_n, a). \end{cases} \tag{7}$$

Similarly, we evaluate all possible thoughts $\mathcal{T}_n$ for state s_n. The quality of a thought t_n is assessed by the average advantage values of all actions guided by it. This allows us to construct the most preferred thought t_n^w and the least preferred thought t_n^l:

$$\begin{cases} t_n^w = \arg\max_{t \in \mathcal{T}_n} \frac{1}{|\mathcal{A}_n|} \sum_{a \in \mathcal{A}_n} A(s_n, a), \\ t_n^l = \arg\min_{t \in \mathcal{T}_n} \frac{1}{|\mathcal{A}_n|} \sum_{a \in \mathcal{A}_n} A(s_n, a). \end{cases} \tag{8}$$

Using these preferences, we construct a dataset $\mathcal{D}$ of tuples containing preferred and dispreferred actions. The Actor is fine-tuned by minimizing the DPO loss function:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(s_n, a_n^w, a_n^l) \sim \mathcal{D}} \ell(\pi_\theta, \pi_{\text{ref}}, s_n, a_n^w, a_n^l),$$
$$\text{with } \ell(\cdot) = \log \sigma[\beta \log(\frac{\pi_\theta(a_n^w|s_n)}{\pi_{ref}(a_n^w|s_n)}) - \beta \log(\frac{\pi_\theta(a_n^l|s_n)}{\pi_{ref}(a_n^l|s_n)})]. \tag{9}$$

Here, the policy π_θ is adapted from the reference LLM π_{ref} through fine-tuning. The Advisor is optimized similarly, but we omit its details for brevity.

Finally, to improve the accuracy of the Critic's benefit estimation, we optimize its model using a temporal difference approach:

$$V(s_n) = r_n + \gamma V(s_{n+1}). \tag{10}$$

4.3 Reward Functions

To compute the advantage value in Eq. (6), we design three reward functions:

Coherence measures the logical progression of recommendations toward the target item. To ensure smooth transitions, we calculate the L2 distance between embeddings of consecutive items, a_{n-1} and a_n, using Llama-3.1-8b:

$$R_C = \|e_{a_{n-1}}, e_{a_n}\|, \tag{11}$$

where $\|\cdot\|$ denotes the L2 distance.

Distance to the target quantifies progress toward the target item i_{Ta}. We compute the difference in L2 distances between a_{n-1} and a_n relative to i_{Ta}:

$$R_T = \|e_{a_{n-1}}, e_{i_{Ta}}\| - \|e_{a_n}, e_{i_{Ta}}\|. \tag{12}$$

User acceptance evaluates user engagement with recommendations. Following (Gao et al., 2023a), we use a simulated environment trained on offline data to provide feedback:

$$R_A = \text{Simulator}(a_n). \tag{13}$$

The final reward is a weighted sum of the three components, provided as feedback to the Actor and Advisor. Each reward's significance is also communicated via prompts, enabling the Advisor to analyze and optimize recommendations:

$$r_n = \alpha R_C + \beta R_T + \delta R_A. \tag{14}$$

5 Experiments

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

RQ1. How does our T-PRA framework compare to SOTA approaches and other baselines in terms of performance?

RQ2. How do the hyperparameters in our method influence the experimental results?

RQ3. What is the performance of our method when given different objective configurations and modeling scenarios?

RQ4. What is the contribution of each component of T-PRA?

5.1 Dataset

We conduct the experiment on two real-world datasets, Steam dataset and Amazon-Book dataset. The Steam dataset contains 190,365 users interacting with 6,012 video games, resulting in a total of 18,523 user interaction sequences after omitting those shorter than 20 interactions. Similarly, the Amazon-Book dataset includes 3,109 users and 13,864 books, culminating in 3,374 user interaction sequences excluding sequences shorter than 20. In our study, the target items are randomly sampled from all items within each respective dataset. For the purpose of training and evaluation, the datasets are separated into training and testing sets, with proportions of 80% and 20%, respectively.

5.2 Metrics

To comprehensively assess the methods, we employ both LLM-based and traditional metrics. LLM-based metrics leverage autonomous LLMs to simulate users to assess the effectiveness of the recommendations. Conversely, traditional metrics are based on a simulator that utilizes the Transformer model (Kang and McAuley, 2018) to evaluate the performance.

Traditional metrics. According to prior research (Bi et al., 2024a; Zhu et al., 2023; Wang et al., 2025a), we employ Increment of Interest (IoI) and Increment of Rank (IoR) as experimental metrics to assess the enhancement of user interest in target items. These metrics are evaluated by training an independent next-item recommender system (Kang and McAuley, 2018) that works as a user simulator for evaluation. Additionally, based on the same simulator, we employ the metric *Accuracy* to measure the mean accuracy of recommended items that align with user preferences:

$$\text{Accuracy} = \frac{1}{|\tau|}\sum_{n=1}^{|\tau|}(1 \text{ if } R(a_n|s_{n-1}) < \theta, \text{else } 0), \tag{15}$$

where $R(\cdot)$ denotes the rank of item a_n generated by the user simulator based on s_{n-1}, θ denotes the threshold of user preference. If the rank is lower than θ, it means the user prefers the item a_n.

LLM-based metrics. We implement metrics *Coherence* and *Acceptance* in prior work Wang et al. (2025a) to appropriately evaluate the LLM-based method, as LLMs are more capable of revealing the latent relationships between items than traditional metrics. The prompt can be found in Appendix A.2.

5.3 Baselines

We implement the following works as the baselines. For adapting the traditional recommendation method to the PR tasks such as **Caser** (Tang and Wang, 2018), **GRU4Rec** (Jannach and Ludewig, 2017), and **SASRec** (Kang and McAuley, 2018), we use greedy search algorithm to select the most relevant items to the target item within the top-k

		Steam					Amazon book				
		IoI	IoR	Acce.	Coh.	Acc.	IoI	IoR	Acce.	Coh.	Acc.
Optimized Traditional RS	POP	-0.426	41.8	0.381	0.482	0.097	0.494	-205.7	0.428	0.595	0.085
	Caser	0.269	89.9	0.505	0.239	**0.977**	0.263	317.1	0.446	0.629	**0.969**
	GRU4Rec	0.220	60.8	0.511	0.240	0.969	0.672	725.6	0.443	**0.645**	0.956
	SASRec	0.354	-75.14	0.259	0.257	0.484	1.06	436.8	0.438	0.506	0.563
	BiLLP	0.427	308.7	0.523	0.477	0.855	1.200	655.3	0.583	0.604	0.490
Proactive RS	IRS	0.164	218.7	0.381	0.249	0.934	0.097	166.4	0.470	0.481	0.883
	LLM-IPP	0.259	340.6	**0.651**	**0.597**	0.912	1.436	845.6	0.595	0.557	0.844
	LLM-IPP (CoT)	0.264	303.0	0.629	0.580	0.906	1.277	803.5	**0.601**	0.538	0.836
	LLM-IPP (ToT)	0.282	244.1	0.571	0.509	0.861	0.944	513.0	0.522	0.533	0.765
	Ours	**0.584**	**432.8**	0.588	0.403	0.894	**1.783**	**1276.5**	0.589	0.629	0.773

Table 1: Results of all methods in two datasets (Bold: Best, Underline: Runner-up).

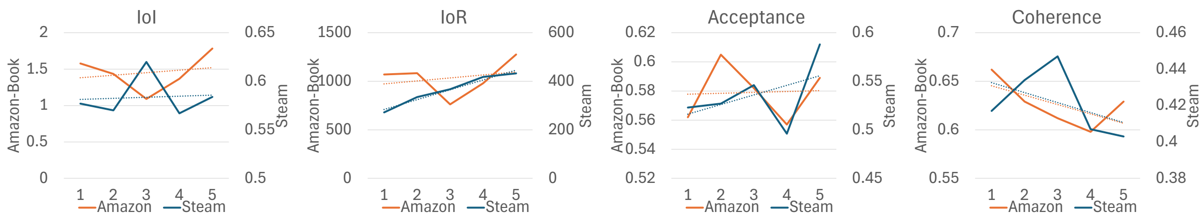

Figure 4: Results of different learning epochs. The dashed line illustrates the trend of values.

recommendations. These items are then aggregated to formulate the final proactive recommendation strategy. For other baselines: **POP** recommends the most popular items; **IRS** (Zhu et al., 2023) adapt a Transformer-based framework to proactive recommendation; **BiLLP** (Shi et al., 2024) employs a bi-level planning approach aimed at augmenting long-term user engagement; **LLM-IPP** (Wang et al., 2025a) utilize the LLMs for proactive recommendation based on prompt engineering techniques such as Chain-of-Thought (CoT) and Tree-of-Thought (ToT).

5.4 Implementation Details

In this study, we employ the Llama-3.1-8b-instruct (Dubey et al., 2024) model as the base model of all the agents in T-PRA. The prompts can be found in Appendix A.1. During the agent tuning process, these models trained for 5 epochs per dataset. The training employs a learning rate of 5e-5 and utilizes a cosine scheduler with a warm-up ratio of 10% and 8 gradient accumulation steps. For the LoRA settings, we set the LoRA rank as 8, and train all modules. All experiments are conducted using 2 NVIDIA A40 GPUs.

5.5 Experiment Results

5.5.1 Main Results (RQ1)

In this section, we evaluate all methods in Steam and Amazon-Book datasets. According to the results presented in Table 1, our proposed method

	Steam				
	IoI	IoR	Acce.	Coh.	Acc.
Ours	0.584	432.8	0.588	0.403	0.894
LLM-IPP	0.422	317.5	0.575	0.336	0.851
	Amazon-Book				
Ours	1.783	1276.5	0.589	0.629	0.773
LLM-IPP	1.308	704.2	0.587	0.423	0.747

Table 2: Comparison between SOTA and T-PRA with the same base model (Llama-3.1-8b-instruct).

outperforms all baseline methods in IoI and IoR. This outcome demonstrates that T-PRA effectively enhances user interest in the target item, thereby achieving the objective of exploring user interests. Regarding user acceptance and coherence, our method achieves comparable results to the SOTA on the Amazon-Book dataset. However, on the Steam dataset, our method performs inferiorly to SOTA. This discrepancy may be attributed to a trade-off between user acceptance and the breadth of user interest, suggesting that although IoR and IoI improvements are notable, they may adversely affect user acceptance and the coherence of the recommendations. Particularly for the Caser and GRU4Rec methods, both exhibit high levels of user acceptance. However, they do not expand user interests at all. Additionally, we conduct a case study to demonstrate the performance of T-PRA at Appendix C.

Furthermore, our experiments utilize the Llama-3.1-8b-instruct as the base model, whereas the SOTA method LLM-IPP uses GPT-3.5-turbo. This

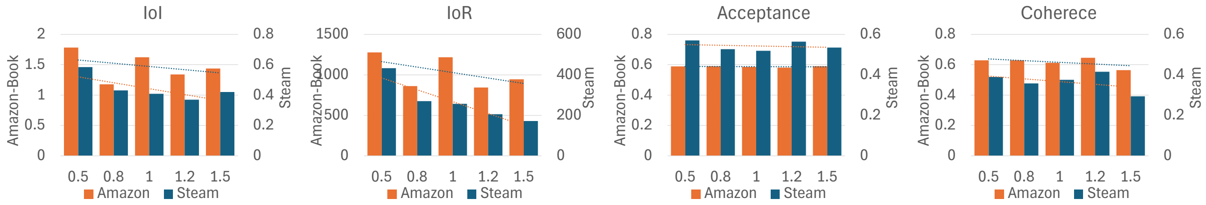

Figure 5: Results of different temperatures. The dashed line illustrates the trend of values.

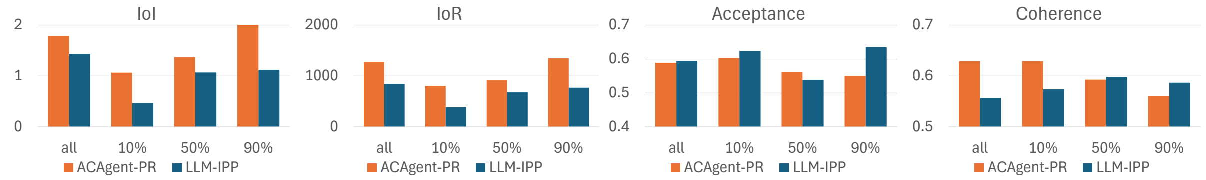

Figure 6: Results of different target item distribution.

History Length	Method	Amazon-Book				
		IoI	IoR	Acce.	Coh.	Acc.
10	Ours	2.293	1320.3	0.521	0.572	0.650
	LLM-IPP	1.444	614.0	0.577	0.492	0.809
15	Ours	1.783	1276.5	0.589	0.629	0.773
	LLM-IPP	1.436	845.6	0.595	0.557	0.844
20	Ours	0.875	566.8	0.567	0.580	0.659
	LLM-IPP	0.604	285.4	0.522	0.485	0.829

Table 3: Results of different history lengths.

	Amazon-Book				
	IoI	IoR	Acce.	Coh.	Acc.
all	1.783	1276.5	0.589	0.629	0.773
only R_C	**1.310**	906.4	0.573	**0.599**	0.755
only R_T	1.276	836.5	0.516	0.619	0.674
only R_A	1.257	**914.0**	**0.597**	0.592	**0.834**

Table 4: Results of different reward objectives.

presents inherent performance differences between the models. We provide a comparison of LLM-IPP's performance using the same model in Table 2, where our approach exceeds LLM-IPP across all metrics.

5.5.2 Hyperparameter Tuning (RQ2)

In this section, we conduct hyperparameter tuning studies to evaluate the impact of hyperparameters of our framework. *Training Epochs* refer to the number of iterations for agent tuning, which can further enhance the effectiveness of agent tuning by repeatedly obtaining preference data from the training set. *Temperature* is a parameter for LLM generation that controls the probability distribution of the next token predicted by the LLMs. A higher temperature can provide more random actions, thereby obtaining a preference dataset with more obvious differences, which, in turn, influences the effectiveness of agent tuning.

Figure 4 indicates that more training epochs lead to a noticeable upward trend in both IoI and IoR. Meanwhile, there is no significant change in user acceptance and strategy coherence. This demonstrates that proactive recommendations can enhance performance through further training without adversely affecting user experience.

From Figure 5, it is observed that our method achieves optimal performance at a temperature setting of 0.5. As the temperature increases, the recommendation performance decreases. This might be because although a higher temperature produces a dataset with more distinct preference comparisons, the increased randomness also reduces the quality of generated actions, which consequently decreases the effectiveness of the agent tuning.

From Table 3, our method consistently outperforms the SOTA across various history lengths. We observed that both our method and LLM-IPP experience a significant performance decline when handling long histories. This decline might be because the outdated history fails to accurately reflect current user interests, resulting in recommendations that diverge from actual user preferences. From the table, it is evident that LLM-IPP suffers particularly severe degradation. For our method, initially, in training epoch 1, our method had an IoI of 0.6 and an IoR of 142, which were lower than LLM-IPP. However, through iterative agent learning, our framework demonstrated substantial improvements compared to LLM-IPP, underscoring the effectiveness of our framework.

5.5.3 Effect on Objectives (RQ3)

In this section, we study the impact of different objective configurations of our framework on Amazon-Book dataset. First, the differences in the reward function enable the agent to optimize the model for different reward targets. Next, we conduct experiments on the distribution differences of target items. Based on the distance between user's historical record and all items, we divide the data into three distributions: the top 10%, 45-55%, and beyond 90%. The target items are then sampled from each distribution for evaluation.

Table 4 indicates that employing any single reward objective cannot achieve optimal outcomes. It's necessary to combine the three reward functions to enhance the proactive recommendation performance. Additionally, it is observed that utilizing R_A as the only objective for training is more effective in increasing user acceptance compared to the other two cases, indicating the effectiveness of reward function.

According to Figure 6, our approach consistently outperforms the SOTA baseline across various target item distributions. It is also observed that when the target item closely aligns with user's historical data, which means the user might be interested in the item and cannot increase the interest too much, resulting in limited enhancement of IoI and IoR. And It's obvious that user acceptance and overall coherence are higher than in the other two cases.

5.5.4 Ablation Study (RQ4)

In this section, we study the contribution of each component of T-PRA on Amazon-Book dataset. Since the Actor is essential for generating recommendations, making ablation experiments is impractical. We conducted experiments on the Advisor and Critic. As observed from the Table 5, the absence of either the Advisor or Critic leads to a decline in performance. Notably, without the Advisor, the initial performance at epoch 1 was reasonable, with IoR at 1065.7 and IoI at 1.486. However, upon agent learning, the performance gradually decreases unlike others, even falling below method without both Advisor and Critic. This indicates the Advisor is a critical role in agent tuning. For the method without the Critic, accurate evaluation of long-term benefits is impeded, resulting in significantly lower performance compared to our proposed method.

	Amazon-Book				
	IoI	IoR	Acce.	Coh.	Acc.
all	1.783	1276.5	0.589	0.629	0.773
no Critic	1.608	883.5	0.501	0.554	0.736
no Advisor	0.774	546.5	0.567	0.546	0.734
without both	1.533	563.0	0.561	0.492	0.681

Table 5: Results of ablation study.

6 Conclusion

This paper proposed the Tunable LLM-based Proactive Recommendation Agent (T-PRA), which can effectively improve proactive recommendation performance by being flexible to user feedback and optimizing long-term overall rewards. We first implement an LLM-based Agent framework to reflect user feedback then improve the recommendation in real time. After that, we utilize agent tuning methods based on A2C algorithm and DPO finetune techniques to train the agent how to recommend items with high long-term rewards. Experimental results demonstrate an average increase of 38% in effectiveness to enhance user interest in a particular item compared to the SOTA methods.

Limitations

Our proposed framework has the following limitations and corresponding future research directions. Initially, regarding agent tuning, our method needs to generate multiple actions at each step for comparison, which is time-intensive. Thus, we can find more efficient agent tuning methods. Additionally, proactive recommendation systems frequently encounter biases, such as those between actual user needs and evaluation metrics (Gao et al., 2025a,b), as well as the reward function configuration. Future work could involve employing online evaluation and inverse reinforcement learning to address these issues. Furthermore, we can apply our proposed approach to other tasks, such as personalized generation, where our framework could finetune LLMs to enhance output with higher long-term user satisfaction and experience.

Acknowledgements

We are deeply grateful to Wentao Shi for his insightful guidance and constructive suggestions regarding the methodology employed in this study. This work is supported by the National Key Research and Development Program of China (2022YFB3104701), the National Natural Science Foundation of China (62272437, U21B2026 and 62402470).

References

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 475–484, New York, NY, USA. Association for Computing Machinery.

Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long- and short-term user representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL '19, pages 336–345, Florence, Italy. Association for Computational Linguistics.

Qazi Mohammad Areeb, Mohammad Nadeem, Shahab Saquib Sohail, Raza Imam, Faiyaz Doctor, Yassine Himeur, Amir Hussain, and Abbes Amira. 2023. Filter bubbles in recommender systems: Fact or fallacy—a systematic review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(6):e1512.

Keqin Bao, Jizhi Zhang, Yang Zhang, Xinyue Huo, Chong Chen, and Fuli Feng. 2024. Decoding matters: Addressing amplification bias and homogeneity issue for llm-based recommendation. *EMNLP*.

Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. 2018. Content-based citation recommendation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL '18, pages 238–251, New Orleans, Louisiana. Association for Computational Linguistics.

Shuxian Bi, Wenjie Wang, Hang Pan, Fuli Feng, and Xiangnan He. 2024a. Proactive recommendation with iterative preference guidance. In *WWW*, page 871–874. Association for Computing Machinery.

Shuxian Bi, Wenjie Wang, Hang Pan, Fuli Feng, and Xiangnan He. 2024b. Proactive recommendation with iterative preference guidance. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 871–874, New York, NY, USA. Association for Computing Machinery.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. Chateval: Towards better LLM-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*.

Zhiyu Chen, Bing Liu, Seungwhan Moon, Chinnadhurai Sankar, Paul Crook, and William Yang Wang. 2022. KETOD: Knowledge-enriched task-oriented dialogue. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2581–2593, Seattle, United States. Association for Computational Linguistics.

Hao Cheng, Shuo Wang, Wensheng Lu, Wei Zhang, Mingyang Zhou, Kezhong Lu, and Hao Liao. 2023. Explainable recommendation with personalized review retrieval and aspect learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, ACL '23, pages 51–64, Toronto, Canada. Association for Computational Linguistics.

Boyi Deng, Wenjie Wang, Fengbin Zhu, Qifan Wang, and Fuli Feng. 2024. Cram: Credibility-aware attention modification in llms for combating misinformation in rag. *arXiv preprint arXiv:2406.11497*.

Yang Deng, Wenqiang Lei, Wenxuan Zhang, Wai Lam, and Tat-Seng Chua. 2022. PACIFIC: Towards proactive conversational question answering over tabular and textual data in finance. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6970–6984, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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. *Preprint*, arXiv:2407.21783.

Chongming Gao, Ruijun Chen, Shuai Yuan, Kexin Huang, Yuanqing Yu, and Xiangnan He. 2025a. Sprec: Self-play to debias llm-based recommendation. In *Proceedings of the ACM Web Conference 2025*, WWW 2025.

Chongming Gao, Mengyao Gao, Chenxiao Fan, Shuai Yuan, Wentao Shi, and Xiangnan He. 2025b. Process-supervised llm recommenders via flow-guided tuning. In *Proceedings of the 48th international ACM SIGIR conference on research and development in information retrieval*, SIGIR 2025.

Chongming Gao, Kexin Huang, Jiawei Chen, Yuan Zhang, Biao Li, Peng Jiang, Shiqi Wang, Zhong Zhang, and Xiangnan He. 2023a. Alleviating matthew effect of offline reinforcement learning in interactive recommendation. In *SIGIR*.

Chongming Gao, Shiqi Wang, Shijun Li, Jiawei Chen, Xiangnan He, Wenqiang Lei, Biao Li, Yuan Zhang, and Peng Jiang. 2023b. Cirs: Bursting filter bubbles by counterfactual interactive recommender system. *ACM Transactions on Information Systems (TOIS)*, 42(1).

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173, Singapore. Association for Computational Linguistics.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, Brussels, Belgium. Association for Computational Linguistics.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 306–310, New York, NY, USA. Association for Computing Machinery.

Daniel Kahneman. 2011. Thinking, fast and slow. *Farrar, Straus and Giroux*.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*, pages 197–206.

Hyunwoo Kim, Youngjae Yu, Liwei Jiang, Ximing Lu, Daniel Khashabi, Gunhee Kim, Yejin Choi, and Maarten Sap. 2022. ProsocialDialog: A prosocial backbone for conversational agents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4005–4029, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35 of *NeurIPS '22*, pages 22199–22213.

Rachna Konigari, Saurabh Ramola, Vijay Vardhan Alluri, and Manish Shrivastava. 2021. Topic shift detection for mixed initiative response. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 161–166, Singapore and Online. Association for Computational Linguistics.

Agnieszka Lawrynowicz, Nicole Merkle, and Ralf Mikut. 2024. Context-aware composition of agent policies by markov decision process entity embeddings and agent ensembles. *Semantic Web*, 15(4):1443–1471.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Yahong Lian, Chunyao Song, and Tingjian Ge. 2025a. ITMPRec: Intention-based targeted multi-round proactive recommendation. In *THE WEB CONFERENCE 2025*, WWW '25.

Yahong Lian, Chunyao Song, and Tingjian Ge. 2025b. Itmprec: Intention-based targeted multi-round proactive recommendation. In *Companion Proceedings of the ACM Web Conference 2025*, WWW '25. Association for Computing Machinery.

John Mendonça, Isabel Trancoso, and Alon Lavie. 2024. Soda-eval: Open-domain dialogue evaluation in the age of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11687–11708, Miami, Florida, USA. Association for Computational Linguistics.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA. PMLR.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Oscar J. Romero, John Zimmerman, Aaron Steinfeld, and Anthony Tomasic. 2024. Synergistic integration of large language models and cognitive architectures for robust ai: An exploratory analysis. *Proceedings of the AAAI Symposium Series*, 2(1):396–405.

Han Shao, Lee Cohen, Avrim Blum, Yishay Mansour, Aadirupa Saha, and Matthew Walter. 2023. Eliciting user preferences for personalized multi-objective decision making through comparative feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 12192–12221. Curran Associates, Inc.

Wentao Shi, Xiangnan He, Yang Zhang, Chongming Gao, Xinyue Li, Jizhi Zhang, Qifan Wang, and Fuli Feng. 2024. Large language models are learnable planners for long-term recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 1893–1903, New York, NY, USA. Association for Computing Machinery.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc.

Samriddhi Singla, Ahmed Eldawy, Tina Diao, Ayan Mukhopadhyay, and Elia Scudiero. 2021. Experimental study of big raster and vector database systems. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2243–2248.

Kai Sun, Seungwhan Moon, Paul Crook, Stephen Roller, Becka Silvert, Bing Liu, Zhiguang Wang, Honglei Liu, Eunjoon Cho, and Claire Cardie. 2021. Adding chit-chat to enhance task-oriented dialogues. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1570–1583, Online. Association for Computational Linguistics.

Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 565–573, New York, NY, USA. Association for Computing Machinery.

Jianling Wang, Haokai Lu, Yifan Liu, He Ma, Yueqi Wang, Yang Gu, Shuzhou Zhang, Ningren Han, Shuchao Bi, Lexi Baugher, Ed H. Chi, and Minmin Chen. 2024. Llms for user interest exploration in large-scale recommendation systems. In *Proceedings of the 18th ACM Conference on Recommender Systems*, RecSys '24, page 872–877, New York, NY, USA.

Mingze Wang, Shuxian Bi, Chongming Gao, Wenjie Wang, Yangyang Li, and Feng Fuli. 2025a. Leveraging llms for influence path planning in proactive recommendation. In *Companion Proceedings of the ACM Web Conference 2025*, WWW '25. Association for Computing Machinery.

Mingze Wang, Shuxian Bi, Wenjie Wang, Chongming Gao, Yangyang Li, and Fuli Feng. 2025b. Incorporate llms with influential recommender system. In *International World Wide Web Conference*, WWW '25.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, Yitao Liang, and Team CraftJarvis. 2023. Describe, explain, plan and select: interactive planning with large language models enables open-world multi-task agents. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Huiyuan Xie, Zhenghao Liu, Chenyan Xiong, Zhiyuan Liu, and Ann Copestake. 2021. TIAGE: A benchmark for topic-shift aware dialog modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1684–1690, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2021. Bot-adversarial dialogue for safe conversational agents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2950–2968, Online. Association for Computational Linguistics.

Yiyan Xu, Jinghao Zhang, Alireza Salemi, Xinting Hu, Wenjie Wang, Fuli Feng, Hamed Zamani, Xiangnan He, and Tat-Seng Chua. 2025. Personalized generation in large model era: A survey. *Preprint*, arXiv:2503.02614.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36 of *NeurIPS '23*, pages 11809–11822. Curran Associates, Inc.

Danyang Zhang, Lu Chen, Situo Zhang, Hongshen Xu, Zihan Zhao, and Kai Yu. 2023. Large language models are semi-parametric reinforcement learning agents. In *Advances in Neural Information Processing Systems*, volume 36, pages 78227–78239. Curran Associates, Inc.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19632–19642.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2024. Progressive-hint prompting improves reasoning in large language models. In *AI for Math Workshop ICML 2024*.

Yiheng Zhou, Yulia Tsvetkov, Alan W Black, and Zhou Yu. 2020. Augmenting non-collaborative dialog systems with explicit semantic and strategic dialog history. In *International Conference on Learning Representations*.

H. Zhu, H. Ge, X. Gu, P. Zhao, and D. Lee. 2023. Influential recommender system. In *ICDE*, pages 1406–1419. IEEE Computer Society.

A LLM Prompts

A.1 Advisor, Actor & Critic

Advisor and Actor use the same prompt shown below, generating thoughts or actions based on the interaction history.

Prompt for Advisor and Actor

You are a helpful planner on recommendation tasks to influence people's interests toward the target item. Given the user's historical data and the target item, analyze the user's interests and finish the recommendation task. You should solve the recommendation task with interleaving Thought, Action, and Observation steps.

- Thought can reason about the current situation and current user interest to help solve the question. Don't give the same Thoughts multiple times.
- Action can recommend an item to the user based on the user's interest and the target item. Your goal is to recommend a sequence of items one by one and finally let the user be interested in the target item step by step. Try to recommend the items the user might like, and any adjacent recommended items should have a strong relation. DO NOT recommend the items that appeared before. If the step is more than 10 and the user's interest is very near the target item, then you can recommend the target item directly. Action must be in the format of: "recommend[item]".

- Observation shows the user's feedback in the following metrics:
- User preference score: indicates the user's preference for this recommended item. The score ranges from 0 to 1. 0 means the user is definitely not interested in the recommended item, and 1 means the user is definitely interested in it.
- Step length reward: indicate this and the last recommended item's distance. Range from 0 to 1, 0 means the recommended item is too different or too same as the last recommended item. 1 means the recommended item has a proper distance from the last recommended item.
- Forward length reward: indicate whether the recommended item is closer to the target item. Range from -1 to 1. Under 0 means that the recommended item is far away from the target item. Above 0 means that the recommended item is closer to the target item.

A.2 LLM-based evaluator

According to the prior work (Wang et al., 2025a), the prompts shown below are LLM prompts to measure user acceptance and coherence of the recommendation sequences based on the recommendations.

Prompt for measuring user acceptability

Given the historical data, analyze the user's interest. Based on this information, would the user be interested in the movies in the proactive recommendation strategy step by step? Answer with a probability for each movie between 0 and 1, where 0 means "definitely not interested" and 1 means "definitely interested". Please explain the reason for each score. If uncertain, make your best guess.

Prompt for measuring coherence

Given the influence path in the format of A,B,C..., what's the relevance of each adjacent item? Answer with a score between 0 and 1, where 0 means "definitely not related" and 1 means "definitely related". Please explain the reason for each score. If uncertain, make your best guess.

B Additional Experiment Results

This section presents all the experimental results in Figures 4, 5, and 6, corresponding to Tables 6, 7, and 8, respectively.

	Steam					Amazon book				
Epoch	IoI	IoR	Acce.	Coh.	Acc.	IoI	IoR	Acce.	Coh.	Acc.
1	0.577	272.0	0.523	0.417	0.864	1.578	1069.8	0.562	**0.662**	0.781
2	0.570	334.7	0.527	0.434	0.870	1.434	1083.3	**0.605**	0.629	**0.817**
3	**0.620**	367.5	0.546	**0.447**	0.881	1.091	761.52	0.582	0.612	0.781
4	0.567	417.9	0.496	0.407	0.866	1.368	979.48	0.557	0.598	0.751
5	0.584	**432.8**	**0.588**	0.403	**0.894**	**1.783**	**1276.5**	0.589	0.629	0.773

Table 6: Results of different training epochs.

	Steam					Amazon book				
Temp.	IoI	IoR	Acce.	Coh.	Acc.	IoI	IoR	Acce.	Coh.	Acc.
0.5	**0.584**	**432.8**	**0.570**	**0.390**	0.894	**1.783**	**1276.5**	0.589	0.629	0.773
0.8	0.431	269.9	0.527	0.358	0.886	1.178	861.0	**0.591**	0.629	0.791
1.0	0.409	256.6	0.519	0.376	**0.902**	1.623	1217.5	0.586	0.613	**0.793**
1.2	0.370	205.6	0.564	0.415	0.859	1.340	844.74	0.581	**0.646**	0.785
1.5	0.420	171.8	0.535	0.294	0.805	1.438	945.5	**0.591**	0.564	0.755

Table 7: Results of different temperatures.

		Amazon-Book				
		IoI	IoR	Acce.	Coh.	Acc.
all	T-PRA	**1.783**	**1276.5**	0.589	**0.629**	0.773
	LLM-IPP	1.436	845.6	**0.595**	0.557	**0.844**
top 10%	T-PRA	**1.065**	**804.5**	0.603	**0.629**	0.801
	LLM-IPP	0.470	384.8	**0.624**	0.574	**0.855**
45-55%	T-PRA	**1.373**	**916.2**	**0.561**	0.593	0.734
	LLM-IPP	1.071	677.4	0.539	**0.598**	**0.807**
beyond 90%	T-PRA	**2.034**	**1346.7**	**0.550**	0.560	0.806
	LLM-IPP	1.121	770.6	0.6355	**0.587**	**0.832**

Table 8: Results of different target item distributions

C Case Study

From the case study presented below, in the beginning, the Advisor conducts an analysis based on the user's historical interaction behaviors at Thought 1. This thought suggests a preference for exploratory and creative games. Consequently, Actor leads to the recommendation of "Roller Coaster Rampage," based on the guidance and grounded in the item most similar within the dataset. Upon receiving feedback from the environment, the Advisor evaluated the recommendation's effectiveness and underlying reasons based on the three rewards. This analysis facilitated the provision of a recommendation with a higher composite reward. Finally, in step 4, the Advisor determines that the recommended item is closely enough aligned with the target item, and then guides the Actor to recommend the target item directly, thereby accomplishing the proactive recommendation task.

Case Study

The users viewing history is:
LEGOae The Hobbit, Castle Crashersae, Super Mustache, Teeworlds, Greyfox RPG, A grande bagune7a espacial - The big space mess, Syberia II, Team Fortress 2, Platypus II, Train Valley, Lords Of The Fallen, Lup, X-Blades, RollerCoaster Tycoonae 2: Triple Thrill Pack, House of 1,000 Doors: Family Secrets Collectors Edition.
The target item is Euro Truck Simulator.
Please recommend games for this user and try to influence the user to become interested in the target item, and finally recommend the target item. The recommended games must be published before 2018.

Thought 1: Based on the users viewing history, it seems that they enjoy a mix of casual, puzzle, and adventure games with a focus on exploration and creativity. To lead them towards the target item Euro Truck Simulator, I should recommend games that offer open-world exploration, simulation, and a sense of freedom.
Action 1: recommend[Roller Coaster Rampage].
Observation 1: [RollerCoaster Tycoon] can not be recommended, instead, recommend[Roller Coaster Rampage].
Observation: The episode continues. User preference score=0.20, step length reward=0.01, length toward to the target=0.44. Final reward=1.09.

Thought 2: Although the users preference score is not high, the step length reward and length toward the target are positive, indicating that Roller Coaster Rampage is a good starting point to lead the user towards Euro Truck Simulator. To continue guiding them, I should recommend more simulation and open-world games that offer a sense of freedom and exploration.
Action 2: recommend[The Sims 3].
Observation 2: [The Sims 2] can not be recommended, instead, recommend[The Sims 3]
Observation: The episode continues. User preference score=1.00, step length reward=0.25, length toward to the target=-0.31. Final reward=1.57.

Thought 3: The users preference score increased significantly after recommending The Sims 3, indicating that they enjoy simulation games. However, the length toward the target is negative, suggesting that The Sims 3 is not as close to Euro Truck Simulator as I would like. To continue guiding the user, I should recommend games that are more closely related to Euro Truck Simulator.
Action 3: recommend[Euro Truck Simulator 2].
Observation: The episode continues. User preference score=1.00, step length reward=0.60, length toward to the target=-0.10. Final reward=2.51.

Thought 4: The users preference score is high, and the step length reward is also positive, indicating that Euro Truck Simulator 2 is a good choice to recommend. Since Euro Truck Simulator 2 is very close to the target item, we can directly recommend the target item Euro Truck Simulator.
Action 4: recommend[Euro Truck Simulator].
Observation: The episode finished. User preference score=0.40, step length reward=0.06, length toward to the target=0.95. Final reward=2.36.