ReRec: Reasoning-Augmented LLM-based Recommendation Assistant via Reinforcement Fine-tuning

Jiani Huang[1], Shijie Wang[1], Liangbo Ning[1], Wenqi Fan[1*], Qing Li[1]
[1]The Hong Kong Polytechnic University
jianihuang01@gmail.com; shijie.wang@connect.polyu.hk;
BigLemon1123@gmail.com;wenqifan03@gmail.com;
qing-prof.li@polyu.edu.hk

Abstract

With the rise of LLMs, there is an increasing need for intelligent recommendation assistants that can handle complex queries and provide personalized, reasoning-driven recommendations. LLM-based recommenders show potential but face challenges in multi-step reasoning, underscoring the need for reasoning-augmented systems. To address this gap, we propose **ReRec**, a novel reinforcement fine-tuning (RFT) framework designed to improve LLM reasoning in complex recommendation tasks. Our framework introduces three key components: (1) *Dual-Graph Enhanced Reward Shaping*, integrating recommendation metrics like NDCG@K with Query Alignment and Preference Alignment Scores to provide fine-grained reward signals for LLM optimization; (2) *Reasoning-aware Advantage Estimation*, which decomposes LLM outputs into reasoning segments and penalizes incorrect steps to enhance reasoning of recommendation; and (3) *Online Curriculum Scheduler*, dynamically assess query difficulty and organize training curriculum to ensure stable learning during RFT. Experiments demonstrate that ReRec outperforms state-of-the-art baselines and preserves core abilities like instruction-following and general knowledge. Our codes are available at https://github.com/jiani-huang/ReRec.

1 Introduction

With the rapid advancement of AI technologies, users now expect more intelligent, context-aware recommendation systems (RecSys) that understand complex, real-time needs and provide personalized suggestions with clear reasoning (Huang et al., 2025c; Zhang et al., 2024). Traditional methods, such as matrix factorization (MF) and graph neural networks (GNNs) (Fan et al., 2019, 2020; Wang et al., 2020), rely on historical data like user ratings or clicks (Chen et al., 2015; Fan et al., 2022), but struggle to process natural language queries that reflect current preferences. As a result, they fail to meet the demand for intelligent recommendation assistants.

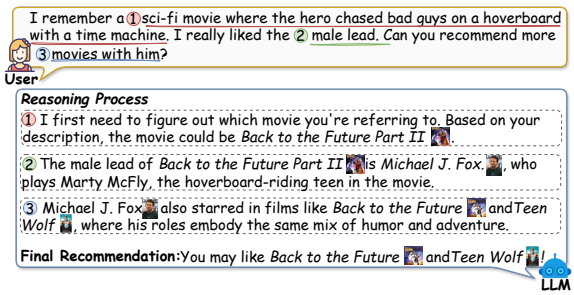

Figure 1: Example of Reasoning-Augmented LLM-based Recommendation Assistant.

The advent of large language models (LLMs) has unlocked new possibilities for intelligent, interactive recommendation assistants (Zhao et al., 2024; Wang et al., 2025c). With their advanced language comprehension, generation abilities, and broad general knowledge (Minaee et al., 2024), LLMs have the potential to understand natural language user queries and generate personalized recommendations. Recent work has demonstrated this potential in developing conversational recommendation systems (CRS) that engage users in multi-turn dialogues before suggesting items (Yang et al., 2024; Liang et al., 2024; Zhu et al., 2025a). However, these dialogues often involve simple and direct user queries (Huang et al., 2025a), such as "Recommend me a sci-fi movie," which require minimal reasoning or constraints. In contrast, user queries often involve more complex queries that demand deeper reasoning for effective decision-making (Ren et al., 2024; Wang et al., 2025b). Consider the complex query illustrated in Figure 1, the assistant must first infer the correct movie from the user's description, then identify the lead actor, and finally suggest other films featuring that ac-

*Corresponding author: Wenqi Fan.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 21040–21055

tor. This process requires the model to engage in multi-step reasoning, going beyond basic attribute matching or shallow semantic understanding. Existing LLM-based RecSys are limited in handling such reasoning-intensive queries due to their inadequate capacity for deep, multi-step reasoning (Shi et al., 2024; Tsai et al., 2024). As a result, there is an urgent need for reasoning-augmented LLMs capable of addressing these complex user requests.

Recent advances in reinforcement fine-tuning (RFT) with rule-based rewards have significantly improved LLMs' reasoning and generalization for various tasks (Xie et al., 2025; Ke et al., 2025; Zou et al., 2025). Unlike supervised fine-tuning (SFT), which requires large amounts of labeled data, RFT uses reinforcement learning (RL) to optimize the model through self-exploration. In this process, the LLM generates responses, while a reward model evaluates them, guiding the model to reinforce effective reasoning strategies. RFT offers better generalization and reduces catastrophic forgetting compared to SFT (Chu et al., 2025), as it focuses on active reasoning rather than memorizing input-output patterns.

However, despite its potential, directly applying RFT to train reasoning-augmented LLM-based recommendation assistants for complex queries presents several challenges. One key challenge lies in developing **fine-grained reward models for complex, query-based recommendation tasks**. In general, the reward model in the RFT framework provides feedback on the recommendation quality, directly guiding the policy model updates. Existing studies often rely on task-specific metrics, such as NDCG, as reward signals, which can be overly stringent and sparse. For instance, when the LLM-based policy model generates recommendations that align with the user's query but deviate from the ground truth, it receives the same zero reward, as responses that entirely fail to address the query. Such coarse rewards may potentially reduce the LLM-based policy model's exploration efficiency, ultimately undermining its overall performance. Another challenge is the **lack of supervision for the reasoning process behind the recommendations**. Recent RFT methods such as GRPO (Shao et al., 2024) typically assign a single reward score to the entire response. As all tokens share this unified score, the LLM policy cannot distinguish which specific parts of the reasoning were correct or flawed. This lack of supervision over intermediate reasoning steps makes it difficult for the model to identify and correct errors in its reasoning, such as misinterpreting user needs or overlooking key constraints of expected items. Consequently, the model struggles to improve its reasoning and may generate suboptimal recommendations (Yang et al., 2024; Zhu et al., 2025a).

To address these challenges, we propose a novel RFT-based framework (**ReRec**) for training a reasoning-augmented LLM-based recommendation assistant. In order to deliver **fine-grained reward signals**, we introduce a *Dual-Graph Enhanced Reward Shaping* mechanism, which enriches traditional metrics like NDCG with two additional components: the Query Alignment Score (QAS) and the Preference Alignment Score (PAS). QAS evaluates how well the recommendations satisfy explicit query constraints using an item-attribute graph, while PAS assesses alignment with user preferences based on similarity to target items. For better **supervision of the reasoning process**, we design *Reasoning-aware Advantage Estimation*, which decomposes the recommendation into reasoning steps and penalizes incorrect ones with lower advantages. Additionally, to mitigate the instability often associated with RL, we introduce the *Online Curriculum Scheduler*, which dynamically reorders training data by prioritizing easier queries based on previous epoch performance, ensuring smoother convergence.

In summary, our contributions are:

- We bridge the gap between recommendation and reasoning, enabling reasoning-augmented LLM-based recommendation assistants to understand users' complex queries and provide reasonable, context-aware recommendations.
- We propose a reinforcement fine-tuning framework **ReRec** that better adapts LLMs to recommendation tasks. It aligns RL signals with recommendation goals, improves reasoning feedback, and enhances training stability, enabling more accurate and context-aware recommendations.
- Extensive experiments demonstrate that our method outperforms state-of-the-art baselines. Additionally, it retains strong instruction-following and reasoning capabilities, ensuring versatility for intelligent recommendations.

2 Preliminaries

Problem Statement. Users often express preferences through complex, multifaceted natural lan-

guage queries, which require the recommendation assistant to perform multi-step reasoning beyond simple keyword matching or attribute filtering to understand the user's intent. Formally, given a user's query q, the LLM-based recommendation assistant π_θ generates a response o, which includes both the reasoning process and a recommended item p. The reasoning should explain why p was selected and why other items were excluded, defined as $\pi_\theta(q) = o$.

Reinforcement Fine-tuning (RFT) for Recommendation. To improve the reasoning capabilities of the LLM-based assistant $\pi_\theta(q)$, reinforcement fine-tuning is typically applied to optimize its policy (Wang et al., 2025d,a). Given an input query q, the assistant generates multiple responses $\{o_1, o_2, \ldots, o_G\}$ based on the learned policy π_θ. These responses are then evaluated by a Reward Model, which assigns reward scores r_i. The scores are used to compute advantages A_i, indicating the quality of each response. These advantages guide the LLM policy to optimize toward improved outputs. Details of the components are:

- *Reward Model ($\mathcal{R}$).* For each generated output o_i, a predefined Reward Model $\mathcal{R}$ computes a corresponding reward value r_i, expressed as $r_i = \mathcal{R}(o_i, gt)$ for each o_i with gt as the ground truth. In the context of LLM-based recommendation, Normalized Discounted Cumulative Gain (NDCG) (Järvelin and Kekäläinen, 2002) or Recall (Gunawardana et al., 2012) can be utilized as a reward model to evaluate output quality.
- *Advantage Estimation (A).* Research shows that directly using rewards for gradient estimation in policy optimization often results in high variance and unstable updates due to a lack of reference point (Schulman et al., 2017). To address this, recent studies introduce the advantage value, which compares the actual reward to the expected reward (Arulkumaran et al., 2017; Mehta, 2020). A positive advantage encourages the policy to favor similar actions. For instance, in sampling-based advantage estimation methods (Ahmadian et al., 2024; Hu, 2025), the LLM policy samples multiple responses $\{o_1, o_2, \ldots, o_G\}$ for a query q, treating each response as a trajectory where tokens are actions. An advantage value $A_{i,t}$ is then computed for each token to identify preferred trajectories.
- *Training Objective ($\mathcal{J}(\theta)$).* Based on the computed advantages, the policy is optimized by maximizing following objective (Shao et al., 2024):

$$\mathcal{J}(\theta) = \mathbb{E}_{(q,gt)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} = \left[\frac{1}{N}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\min\left(h_{i,t}(\theta)A_{i,t}, \text{clip}\left(h_{i,t}(\theta), c_l, c_h\right)A_{i,t}\right)\right], \tag{1}$$

where $N = \sum_{i=1}^{G}|o_i|$, $h_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$, the clip function represents the clipped probability ratio, and $c_l = 1-\varepsilon$, $c_h = 1+\varepsilon$.

3 Methodology

3.1 Overview of the Proposed ReRec

We aims to develop a reasoning-augmented LLM-based recommendation assistant with RFT. While RFT has improved LLM reasoning in various tasks, directly applying it to query-based recommendations is challenging due to coarse reward signals and lack of supervision on intermediate reasoning. To address these issues, we propose ReRec, a novel RFT framework for query-based recommendation tasks. As shown in Figure 2, ReRec introduces a *Dual-Graph Enhanced Reward Shaping* mechanism for better reward guidance, and *Reasoning-aware Advantage Estimation* to supervise intermediate reasoning steps and penalize errors. An *Online Curriculum Scheduler* further stabilizes training by dynamically adjusting the curriculum.

3.2 Dual-Graph Enhanced Reward Shaping

Recent studies often use rule-based rewards in RFT to improve LLM reasoning capabilities (Jin et al., 2025; Huang et al., 2025b; Wei et al., 2025; Luo et al., 2025). For recommendation task, metrics like NDCG@K are commonly adopted as reward models. Although these metrics are established proxies for recommendation accuracy, they are unsuitable for direct use in policy optimization for query-based recommendations. Their primary limitation is evaluating only exact matches with ground-truth items, failing to assess recommendation quality comprehensively. For instance, a recommendation meeting key query constraints (e.g., genre or actor) or showing collaborative signals similar to the ground truth should be considered better than a completely unrelated recommendation, even if it does not match exactly. However, coarse-grained metrics like NDCG@K assign both the same zero reward, unable to distinguish meaningful from irrelevant recommendations. This reliance on coarse-grained rewards can destabilize RFT training, impede convergence to an optimal

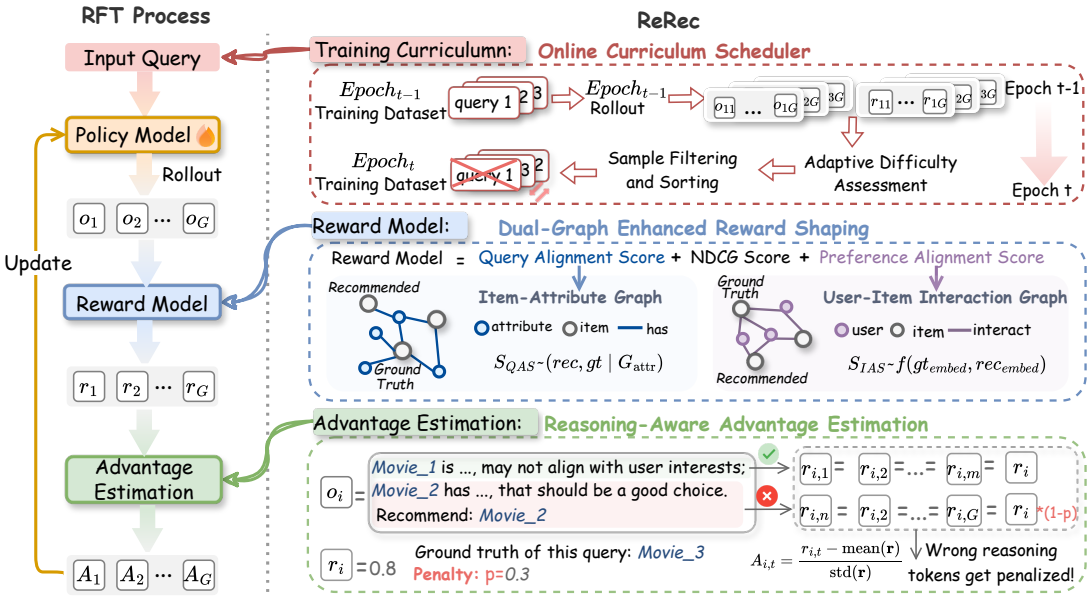

Figure 2: The overall model architecture of the proposed ReRec.

policy, and ultimately impair accurate reasoning and recommendations.

To overcome the above limitations, we introduce a Dual-Graph Enhanced Reward Shaping mechanism that enriches the reward space with complementary fine-grained signals. Specifically, in addition to the recommendation metrics $NDCG@K$, we incorporate below two auxiliary components:

Query Alignment Score (QAS): User queries often specify constraints like genre, actor, or director that recommended items must satisfy. While the ground-truth item is a valid option, other items may also meet these criteria. Relying solely on exact matches with the ground truth for reward assignment overlooks these alternatives, penalizing reasonable recommendations and causing limited exploration of the action space and unstable training. To address this, we leverage item-attribute relationships to better evaluate whether recommended items meet query constraints. Using the ground-truth item as a reference, we assess a recommended item's alignment by comparing shared attributes in the item-attribute graph. For an item p_i predicted by the LLM and ground truth gt, given the item-attribute graph G_{attr}, we compute the QAS as the proportion of shared relationships between p_i and gt, as follows:

$$S_{QAS}(p_i, gt) = \frac{|R_{p_i}^{G_{attr}} \cap R_{gt}^{G_{attr}}|}{|R_{gt}^{G_{attr}}|}. \quad (2)$$

Preference Alignment Score (PAS): While the Query Alignment Score (QAS) evaluates whether a recommended item meets query constraints, it overlooks users' implicit preferences beyond basic attributes. For instance, in the query "movies starring Tom Hanks," users may prefer niche films over blockbusters. Recommending a popular film, though meeting query constraints, should incur a penalty if it misaligns with such preferences. A reward model based solely on attribute matching fails to capture these nuances. To address this, we incorporate collaborative signals from user-item interactions, reflecting co-interaction preferences (He et al., 2017; Sarwar et al., 2001). We pre-train a lightweight recommender model $\mathcal{M}$ (e.g., LightGCN (He et al., 2020)) to generate item embeddings from the user-item interaction graph. The PAS for a recommended item p_i and ground truth gt is defined as the cosine similarity between their embeddings, as follows:

$$S_{PAS}(p_i, gt) = \frac{\mathcal{M}(p_i) \cdot \mathcal{M}(g_t)}{\|\mathcal{M}(p_i)\| \|\mathcal{M}(g_t)\|}. \quad (3)$$

Formally, the final shaped reward integrates all three components and can be expressed as:

$$r_i = \text{NDCG} + w_1 S_{QAS} + w_2 S_{PAS}, \quad (4)$$

where w_1 and w_2 control the influence of the auxiliary scores on the overall reward.

3.3 Reasoning-Aware Advantage Estimation

Existing RFT algorithms with rule-based rewards typically assign the same advantage to all tokens based solely on the final response's reward (Shao

et al., 2024). This causes LLM-based recommendation systems to focus on generating the final answer, neglecting the quality of intermediate reasoning and failing to differentiate correct from incorrect steps. However, the reasoning process is essential for accurate recommendations, especially in complex scenarios that require multi-step reasoning (Qu et al., 2025; Wang et al., 2024a). Assigning the same reward to all tokens prevents the model from identifying flawed reasoning, leading to suboptimal performance. Recent research has explored process reward models to guide LLMs' intermediate reasoning (Choudhury, 2025; Tu et al., 2025). These methods either train dedicated models or use large pre-trained models to score reasoning steps, but both are computationally expensive and face scalability issues, limiting their practicality.

To mitigate these limitations, we propose a lightweight and effective method: *Reasoning-Aware Advantage Estimation* (RAAE). RAAE provides fine-grained supervision of the reasoning process specific to recommendation tasks. Unlike conventional RFT methods, which treat all tokens equally in a reasoning trajectory, RAAE differentiates token-level contributions by penalizing tokens in reasoning steps that lead to incorrect recommendations. Specifically, we decompose the LLM's output into distinct reasoning steps and reward or penalize each segment based on its contribution to the final recommendation.

Mathematically, given a user query q and ground truth gt, the policy of LLM-based recommendation assistant generates an output o_i containing a predicted item p_i and a reasoning process. We decompose o_i into K reasoning steps via paragraph-based segmentation, formalized as:

$$\mathcal{S}_i = \{s_{i,1}, \ldots, s_{i,K}\} \text{ where } \sum_{k=1}^{K} |s_{i,k}| = |o_i|\,, \quad (5)$$

where $s_{i,k}$ represents the k-th reasoning segment of output o_i, each segment contains the reasoning step for one item. We assign rewards to reasoning segments based on whether they involve an incorrectly recommended item. If a segment discusses such an item that is ultimately recommended, it indicates the model failed to exclude it, and the reasoning in that segment is considered incorrect and penalized, as follows:

$$r_{s_{i,k}} = \begin{cases} (1 - w_{penalty}) \cdot r_i & \text{if } (p_i \neq gt) \wedge (p_i \in s_{i,k}), \\ r_i & \text{otherwise}, \end{cases} \quad (6)$$

where $w_{penalty} \in (0, 1)$ is a hyperparameter that penalizes reasoning steps with incorrect predictions, reducing rewards for associated tokens while retaining higher rewards for correct ones. After obtaining reward of each reasoning segment, we map the segment reward $r_{s_{i,k}}$ to each token $t \in s_i^k$ as $r_{i,t}$, forming $\mathbf{r} = \{r_{1,1}, \ldots, r_{G,|o_G|}\}$, and compute the advantage as $A_{i,t} = \frac{r_{i,t} - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$ across all tokens of outputs $\{o_1, \ldots, o_G\}$. The token-level advantage is then used to guide policy optimization by maximizing the objective of Equation (1). This approach enables differential rewards across reasoning steps within the same response, thereby improving the reasoning accuracy and final recommendation performance.

3.4 Online Curriculum Scheduler

Training LLMs for complex recommendation tasks is challenging due to the gap between language generation and recommendation. Early in training, LLMs often struggle with complex queries, resulting in zero reward signals that hinder learning. Curriculum learning, where tasks gradually increase in difficulty, has been proposed as a solution (Narvekar et al., 2020; Narvekar and Stone, 2018; Jiang et al., 2025). However, applying it to recommendation tasks is difficult, as task difficulty is harder to define compared to domains like math or code generation. Difficulty in recommendations depends on factors like the number of constraints, reasoning depth, and user expression variability (Chen et al., 2025). Additionally, traditional curriculum learning methods fail to account for the model's evolving capabilities, as they rely on static difficulty assessments made before training begins (Wang et al., 2024b).

To address these challenges, we propose an *Online Curriculum Scheduler* that dynamically adjusts the training curriculum based on the policy model's evolving capabilities, which consists of three steps:

Adaptive Difficulty Assessment. During RFT training, as the policy improves, queries that were previously difficult may become easier. It is therefore important to adaptively assess query difficulty based on the model's evolving capabilities. This can be done by measuring the model's average performance on each query from the previous epoch. Low rewards indicate that a query is still challenging, while consistently high rewards suggest the query has been mastered and is less challenging in future iterations. Formally, at the start of epoch t,

we evaluate the difficulty of samples from the previous epoch's dataset $\mathcal{D}^{t-1} = q_1, q_2, \ldots, q_N$, where q is a recommendation query. For each q, the model generated G rollout outputs $o_1, o_2, \ldots, o_G$ in epoch $t-1$. The difficulty score d^{t-1} is computed as the average of the inverse rewards across all outputs:

$$d^{t-1} = \frac{1}{G}\sum_{i=1}^{G}(1 - r_i), \tag{7}$$

where r_i is the reward score for the i-th output of q in epoch $t-1$.

Sample Filtering and Sorting. We apply a difficulty threshold τ to filter out "easy" samples where $d^{t-1} < \tau$, as consistent high performance across rollouts suggests minimal learning benefit. The remaining samples are sorted by d^{t-1} in ascending order to form the new dataset $\mathcal{D}^t$:

$$\mathcal{D}^t = \left\{\left(q_{(k)}, d^{t-1}_{(k)}\right)\right\}_{k=1}^{m} \text{ where } \tau \le d^{t-1}_{(1)} \le \cdots \le d^{t-1}_{(m)}. \tag{8}$$

This prioritizes easier samples early in epoch t, fostering stable learning and gradual progression.

Iterative Curriculum Update. The sorted $\mathcal{D}^t$ is used for training in epoch t, and the process repeats for epoch $t+1$ with updated difficulty d^t_n based on the previous rollouts. This dynamic process adapts to the model's evolving abilities while staying efficient, as it reuses existing rollout data without relying on extra models or additional inference.

4 Experiment

We aim to answer the key research questions (RQs):

RQ1. How does ReRec compare to baselines?

RQ2. How effectively can ReRec leverage user interaction history to provide personalized recommendations?

RQ3. How does ReRec perform in generalization, e.g., cross-domain and cross-task settings?

RQ4. To what extent does ReRec retain its original knowledge and capabilities?

4.1 Experiment Setup

4.1.1 Dataset

To evaluate our method, we conducted experiments on RecBench+ (Huang et al., 2025a), a benchmark dataset tailored for assessing complex reasoning in query-based recommendations. It covers two domains (Movie and Book) with user queries divided into five subcategories based on reasoning complexity. Details are provided in **Appendix A**. Data was sampled according to query category distribution, with statistics shown in Table 1.

Table 1: Dataset Statistics

Category	Sub-category	Movie	Book
Condition-based Query	Explicit Condition (Simple)	8,262	10,681
	Implicit Condition (Medium)	5,790	7,741
	Misinformed Condition (Hard)	5,374	7,890
User Profile-based Query	Interest-based	2,365	1,273
	Demographics-based	209	-
Total		22,000	27,585

4.1.2 Baseline Models

We compare our method with three categories of approaches designed to handle such queries effectively: **LLM backbones**, such as Qwen-2.5-3B-Instruct (Team, 2024) and GPT-4o; **LLM-based CRSs**, including TallRec (Bao et al., 2023), InteRecAgent (Huang et al., 2025c), and CRAG (Zhu et al., 2025b); and **RFT-trained Models** like GRPO (Shao et al., 2024), REINFORCE++ (Hu, 2025), and RLOO (Ahmadian et al., 2024), where accuracy is used as the reward during training. See more information about baselines in **Appendix B**.

4.1.3 Evaluation Settings

For each query, we generate a candidate set with 1 positive item and 19 randomly sampled negative items. Models are evaluated on Accuracy, based on selecting the correct item matching the user query.

4.1.4 Implementations

We select Qwen-2.5-3B-Instruct and Llama-3.2-3B-Instruct as the backbone LLM. Due to space limit, more details are provided in **Appendix D.1**.

4.2 Overall Performance (RQ1)

We compare our method with various LLM backbones, LLM-based CRS models, and RFT-trained models using the same test set. The results are presented in Table 2. From the table, it is evident that LLM backbones, such as Qwen-2.5-3B-Instruct and Llama-3.2-3B-Instruct, perform significantly worse than more advanced closed-source models like GPT-4o and DeepSeek-R1. While these LLM backbones excel on simpler tasks, such as Explicit Condition-based Query, their performance significantly declines on more complex tasks requiring deeper reasoning, like Misinformed Condition-based (Hard) Query. LLM-based CRS models achieve slightly better performance across all query types compared to closed-source LLMs. RFT-trained models show the most significant improvement, with ReRec outperforming other RFT-based models across domains and query types. For

Table 2: The overall performance of baselines and ReRec evaluated by Accuracy. The bold/underline values represent the best/the second-best result, respectively.

Category	Model	Movie					Book			
		Simple	Medium	Hard	Interest -based	Demographics -based	Simple	Medium	Hard	Interest -based
LLM Backbone	Qwen-2.5-3B-Instruct	0.284	0.135	0.101	0.369	0.450	0.371	0.304	0.177	0.416
	Llama-3.2-3B-Instruct	0.107	0.052	0.029	0.097	0.193	0.215	0.138	0.106	0.254
	DS-R1-Distill-Qwen-7B	0.083	0.041	0.040	0.133	0.165	0.131	0.104	0.087	0.221
	GPT-4o	0.554	0.519	0.188	0.550	0.504	0.554	0.590	0.160	0.458
	Deepseek-R1	0.537	0.510	0.200	0.459	0.425	0.562	0.530	0.279	0.505
LLM-based CRS	TallRec	0.537	0.533	0.284	0.571	0.509	0.563	0.591	0.251	0.477
	InteRecAgent	0.542	0.529	0.178	0.563	0.548	0.557	0.582	0.147	0.493
	CRAG	0.560	0.531	0.195	0.557	0.513	0.560	0.592	0.211	0.518
RFT-trained Model	Qwen-2.5-3B-Instruct									
	GRPO	0.549	0.502	0.461	0.579	0.648	0.563	0.630	0.552	0.699
	REINFORCE++	0.578	0.523	0.506	0.556	0.637	0.553	0.618	0.510	0.716
	RLOO	0.560	0.495	0.529	0.573	0.614	**0.567**	0.649	0.532	0.716
	ReRec	**0.595**	**0.548**	**0.547**	**0.588**	**0.670**	0.555	**0.655**	**0.562**	**0.746**
	Llama-3.2-3B-Instruct									
	GRPO	0.686	0.600	0.644	0.651	0.642	0.664	0.757	0.713	0.786
	REINFORCE++	0.699	0.623	0.597	0.676	0.771	0.661	0.768	0.697	0.795
	RLOO	0.693	0.609	0.614	0.627	0.688	0.660	0.774	0.704	0.794
	ReRec	**0.748**	**0.700**	**0.729**	**0.719**	**0.800**	**0.671**	**0.782**	**0.750**	**0.811**

example, on the Llama-3.2-3B-Instruct model, our ReRec achieves a performance improvement of 3.76% to 13.2% over the second-best method in the Movie domain. In particular, on more challenging tasks like Misinformed-based (Hard) Query, ReRec demonstrates a remarkable 440% improvement over the untrained model. This significant gain highlights a marked improvement in LLM's reasoning abilities, enabling it to identify misleading information within the query and better understand the user's true intentions.

4.3 Personalized Recommendation (RQ2)

In real-world settings, multiple items may satisfy a given query. To assess the model's ability to personalize recommendations based on user history, we conduct experiments in a personalized scenario. Instead of randomly sampling 19 negative items, we select K = 3 "hard negatives" that meet the query criteria but are not the ground truth. We then compare model performance in two settings: (1) using only the query, and (2) using both the query and the user's interaction history. A model capable of leveraging historical preferences should distinguish the positive item from the hard negatives.

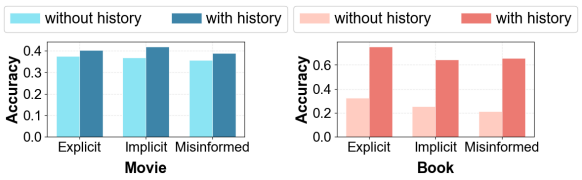

Figure 3: Performance on personalized recommendation

As shown in Figure 3, providing user interaction history improves model performance across different categories and domains. This suggests that the model considers both the query and the user's preferences when selecting items, enabling it to better exclude hard negatives. These results show that ReRec effectively uses user history to reason preferences and generate more personalized recommendations.

4.4 Generalization Capability (RQ3)

The generalization capability is crucial for LLM-based recommendation assistants, as users may request recommendations across a wide range of domains or require the model to perform different types of recommendation tasks. To effectively handle this problem, the assistant needs to generalize well, adapting to new contexts and tasks while maintaining high performance. To explore the generalization capabilities of ReRec, we conducted experiments to assess its cross-domain and cross-task generalization. For **cross-domain generalization**, we evaluated the model's ability to transfer knowledge between domains. For example, a model trained on the Movie domain was applied to zero-shot recommendations on Book data. As shown in Table 3, ReRec (with Llama backbone) achieves a score of 0.494 when trained on Movie and tested on Book, a 181% relative improvement

Table 3: Performance in the generalization (cross-domain) setting. **Bold:** Accuracy on new domain.

Training Dataset	Method	Base Model	Testing Dataset	
			Movie	Book
zero-shot	prompt	Qwen	0.240	0.301
	prompt	Llama	0.078	0.168
	prompt	GPT-4o	0.470	0.453
	prompt	DeepSeek-R1	0.441	0.474
Movie	ReRec	Qwen	0.567	**0.434**
	ReRec	Llama	0.727	**0.494**
Book	ReRec	Qwen	**0.406**	0.598
	ReRec	Llama	**0.448**	0.733

Table 4: Performance on transferring to sequential recommendation.

Model	Accuracy
Llama-3.2-3B-Instruct	0.120
Qwen-2.5-3B-Instruct	0.286
GRU4Rec	0.658
SASRec	**0.673** (*best*)
ReRec-Qwen	0.591 (vs. Qwen +107% | vs. *best* 87.8%)
ReRec-Llama	0.595 (vs. Llama +396% | vs. *best* 88.4%)

over the base model (0.168), outperforming strong models like GPT-4o and DeepSeek-R1. A similar trend was observed when transferring from Book to Movie. These results demonstrate that ReRec effectively generalizes across domains, capturing reasoning patterns that are not domain-specific.

For **cross-task generalization**, we evaluated the model on a different task: sequential recommendation, where it predicts the next item based on a user's interaction history. More details about experiments are provided in **Appendix D.3**. Table 4 indicates that ReRec, trained on our complex query-based task, transfers well to sequential recommendation, achieving substantial gains over base models (Qwen +107%, Llama +396%). Moreover, ReRec achieves up to 90% of the performance of specialized models like SASRec.

4.5 Capability Retention (RQ4)

Beyond recommendation accuracy, a reasoning-augmented LLM-based assistant also needs world knowledge and instruction-following skills to interpret user queries and provide relevant, personalized responses. Preserving the base model's original capabilities after RFT is therefore critical. Prior work has shown that SFT can cause catastrophic forgetting, where task-specific learning leads to loss of previously acquired knowledge (Chu et al., 2025). We evaluate how our method, compared to SFT, maintains the model's original abilities across four

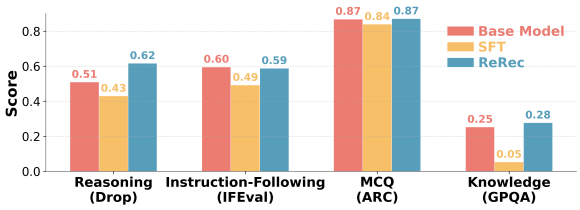

Figure 4: Knowledge and Capability Retention

key dimensions relevant to interactive recommendation: **Reasoning**, **Instruction-Following**, **Multiple Choice Question (MCQ) answering**, and **Knowledge**. Details on the SFT procedure and evaluation benchmarks are given in **Appendix D.4**.

As shown in Figure 4, our model preserves the original model's capabilities across all four dimensions with minimal loss. Remarkably, its reasoning ability improved by 21.6% over the base model. In contrast, the SFT-trained model suffered substantial declines, especially in Reasoning and Knowledge, which dropped by 15.7% and 80%, respectively. Such degradation can harm LLM-based recommendation assistants, making them appear less intelligent and reducing user satisfaction and trust.

4.6 Ablation Study

To evaluate each module's impact, we performed an ablation study by removing components one at a time. As shown in Figure 5, the full model ReRec achieves the highest accuracy. Removing each component reduces performance, while the largest drop occurs when RAEE is removed.

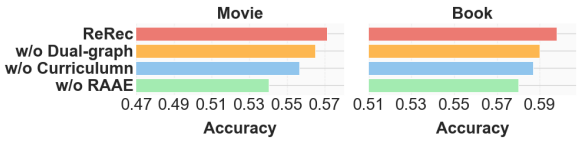

Figure 5: Ablation of ReRec

4.7 Parameter Analysis

In this section, we further explored the influence of the penalty parameter w_{penalty} for incorrect reasoning steps in RAEE on model performance. As shown in Figure 6, accuracy improves as w_{penalty} increases, peaking at 0.30, after which it gradually drops. This suggests that moderate penalization helps suppress incorrect reasoning, while overly strong penalties may harm final performance. Consequently, we set $w_{\text{penalty}} = 0.3$ as the default in this paper to achieve the best performance.

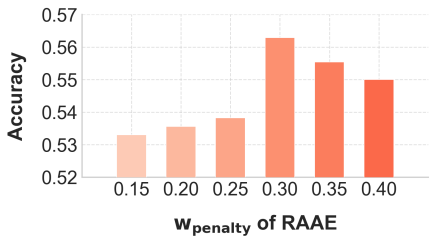

Figure 6: Effect of w_{penalty} of RAAE

5 Related Work

LLM-based Recommendation. With the rapid advancement of LLMs, researchers have increasingly explored their applications in recommender systems (Zhao et al., 2024). For example, Lyu et al. (2023) used LLMs to generate detailed, context-aware user and item profiles from historical interactions, enhancing the expressiveness of input features. TallRec (Bao et al., 2023) applied instruction fine-tuning to LLMs for recommendation tasks. LLMs have also been studied for conversational recommendation systems (CRS). Friedman et al. (2023) proposed RecLLM, a LaMDA-based system for YouTube video recommendation that captures user preferences, manages dialogue flexibly, and produces explainable recommendations. Similarly, Feng et al. (Feng et al., 2023) developed a hybrid architecture combining LLMs with domain-specific expert models to improve CRS performance. Despite these advances, most existing studies focus on short, template-based dialogues with simple intents (e.g., "I want a [genre] movie"), which limits query diversity and ignores the complex reasoning scenarios. To address this, we use RecBench+ (Huang et al., 2025a), a recent benchmark featuring user queries of varying reasoning difficulty to more effectively evaluate recommendation reasoning.

Reinforcement Learning for LLM Reasoning. Recent studies show that reinforcement fine-tuning can greatly enhance the reasoning ability of LLMs. Models such as Deepseek-R1 (Guo et al., 2025) and Kimi K1.5 (Team et al., 2025) use RL algorithms like GRPO (Shao et al., 2024) to enable multi-step reasoning. Several GRPO variants, including DAPO (Yu et al., 2025) and Dr.GRPO (Shao et al., 2024), have further improved the efficiency and effectiveness of RL in post-training. These RL-based methods have been successfully applied in video understanding (Feng et al., 2025; Jiang et al., 2024), audio processing (Rouditchenko et al., 2025), and robotics (Kim et al., 2025), significantly enhancing reasoning and generalization. However, in recommendation domain, RL research has mainly focused on sequential recommendation, with limited attention to reasoning-intensive query-based recommendations.

6 Conclusion

In this paper, we present **ReRec**, a reinforcement fine-tuning framework that enhances LLM-based recommendation assistants with improved reasoning. To better adapt RFT to recommendation tasks, we introduce fine-grained reward shaping and reasoning-aware advantage estimation. Extensive experiments demonstrate that **ReRec** outperforms state-of-the-art baselines in recommendation accuracy, while preserving instruction-following and general knowledge capabilities.

Acknowledgments

The research described in this paper has been partially supported by the General Research Funds from the Hong Kong Research Grants Council (project No. PolyU 15207322, 15200023, 15206024, and 15224524), Hong Kong Research Grants Council's Theme-based Research Scheme (No. T43-513/23-N), Hong Kong Research Grants Council's Research Impact Fund (No. R1015-23), Hong Kong Research Grants Council's Collaborative Research Fund (No. C1043-24GF), Internal research funds from Hong Kong Polytechnic University (project no. P0059586, P0042693, P0048625, and P0051361), and Sheertek International (HK) Limited. This work was supported by computational resources provided by The Centre for Large AI Models (CLAIM) of The Hong Kong Polytechnic University.

Limitations

While our ReRec framework effectively enhances reasoning in single-turn query-based recommendations, it does not account for multi-turn dialogues, which are common in real-world conversational recommendation systems. This may limit its applicability in scenarios requiring ongoing user interactions and context accumulation. Future work could extend the framework to incorporate multi-turn capabilities, such as maintaining conversation history and adapting rewards dynamically across interactions.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*.

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. Deep reinforcement learning: A brief survey. *IEEE signal processing magazine*, 34(6):26–38.

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. In *Proceedings of the 17th ACM conference on recommender systems*, pages 1007–1014.

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2025. Unleashing the potential of prompt engineering for large language models. *Patterns*.

Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*, 25(2):99–154.

Sanjiban Choudhury. 2025. Process reward models for llm agents: Practical framework and directions. *arXiv preprint arXiv:2502.10325*.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*.

Wenqi Fan, Xiaorui Liu, Wei Jin, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2022. Graph trend filtering networks for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 112–121.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426.

Wenqi Fan, Yao Ma, Qing Li, Jianping Wang, Guoyong Cai, Jiliang Tang, and Dawei Yin. 2020. A graph neural network framework for social recommendations. *IEEE Transactions on Knowledge and Data Engineering*.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. 2025. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*.

Yue Feng, Shuchang Liu, Zhenghai Xue, Qingpeng Cai, Lantao Hu, Peng Jiang, Kun Gai, and Fei Sun. 2023. A large language model enhanced conversational recommender system. *arXiv preprint arXiv:2308.06212*.

Luke Friedman, Sameer Ahuja, David Allen, Zhenning Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, and 1 others. 2023. Leveraging large language models in conversational recommender systems. *arXiv preprint arXiv:2305.07961*.

Asela Gunawardana, Guy Shani, and Sivan Yogev. 2012. Evaluating recommender systems. In *Recommender systems handbook*, pages 547–601. Springer.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

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

Jian Hu. 2025. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*.

Jiani Huang, Shijie Wang, Liang-bo Ning, Wenqi Fan, Shuaiqiang Wang, Dawei Yin, and Qing Li. 2025a. Towards next-generation recommender systems: A benchmark for personalized recommendation assistant with llms. *arXiv preprint arXiv:2503.09382*.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025b. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

Xu Huang, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. 2025c. Recommender ai agent: Integrating large language models for interactive recommendations. *ACM Transactions on Information Systems*, 43(4):1–33.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Yiyang Jiang, Guangwu Qian, Jiaxin Wu, Qi Huang, Qing Li, Yongkang Wu, and Xiao-Yong Wei. 2025. Self-paced learning for images of antinuclear antibodies. *IEEE Transactions on Medical Imaging*.

Yiyang Jiang, Wengyu Zhang, Xulu Zhang, Xiao-Yong Wei, Chang Wen Chen, and Qing Li. 2024. Prior knowledge integration via llm encoding and pseudo event regulation for video moment retrieval. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7249–7258.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.

Zixuan Ke, Fangkai Jiao, Yifei Ming, Xuan-Phi Nguyen, Austin Xu, Do Xuan Long, Minzhi Li, Chengwei Qin, Peifeng Wang, Silvio Savarese, and 1 others. 2025. A survey of frontiers in llm reasoning: Inference scaling, learning to reason, and agentic systems. *arXiv preprint arXiv:2504.09037*.

Dongyoung Kim, Sumin Park, Huiwon Jang, Jinwoo Shin, Jaehyung Kim, and Younggyo Seo. 2025. Robot-r1: Reinforcement learning for enhanced embodied reasoning in robotics. *arXiv preprint arXiv:2506.00070*.

Tingting Liang, Chenxin Jin, Lingzhi Wang, Wenqi Fan, Congying Xia, Kai Chen, and Yuyu Yin. 2024. Llm-redial: a large-scale dataset for conversational recommender systems created from user behaviors with llms. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 8926–8939.

Run Luo, Lu Wang, Wanwei He, and Xiaobo Xia. 2025. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Christopher Leung, Jiajie Tang, and Jiebo Luo. 2023. Llm-rec: Personalized recommendation via prompting large language models. *arXiv preprint arXiv:2307.15780*.

Deepanshu Mehta. 2020. State-of-the-art reinforcement learning algorithms. *International Journal of Engineering Research and Technology*, 8(1):717–722.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. 2020. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50.

Sanmit Narvekar and Peter Stone. 2018. Learning curriculum policies for reinforcement learning. *arXiv preprint arXiv:1812.00285*.

Yuxiao Qu, Matthew YR Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. 2025. Optimizing test-time compute via meta reinforcement fine-tuning. *arXiv preprint arXiv:2503.07572*.

Xuhui Ren, Tong Chen, Quoc Viet Hung Nguyen, Lizhen Cui, Zi Huang, and Hongzhi Yin. 2024. Explicit knowledge graph reasoning for conversational recommendation. *ACM Transactions on Intelligent Systems and Technology*, 15(4):1–21.

Andrew Rouditchenko, Saurabhchand Bhati, Edson Araujo, Samuel Thomas, Hilde Kuehne, Rogerio Feris, and James Glass. 2025. Omni-r1: Do you really need audio to fine-tune your audio llm? *arXiv preprint arXiv:2505.09439*.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295.

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

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

Guangsi Shi, Xiaofeng Deng, Linhao Luo, Lijuan Xia, Lei Bao, Bei Ye, Fei Du, Shirui Pan, and Yuxiao Li. 2024. Llm-powered explanations: Unraveling recommendations through subgraph reasoning. *arXiv preprint arXiv:2406.15859*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Qwen Team. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Alicia Y Tsai, Adam Kraft, Long Jin, Chenwei Cai, Anahita Hosseini, Taibai Xu, Zemin Zhang, Lichan Hong, Ed H Chi, and Xinyang Yi. 2024. Leveraging llm reasoning enhances personalized recommender systems. *arXiv preprint arXiv:2408.00802*.

Haoqin Tu, Weitao Feng, Hardy Chen, Hui Liu, Xianfeng Tang, and Cihang Xie. 2025. Vilbench: A suite for vision-language process reward modeling. *arXiv preprint arXiv:2503.20271*.

Jie Wang, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M Jose. 2025a. Large language model driven policy exploration for recommender systems. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 107–116.

Jun Wang, Meng Fang, Ziyu Wan, Muning Wen, Jiachen Zhu, Anjie Liu, Ziqin Gong, Yan Song, Lei Chen, Lionel M Ni, and 1 others. 2024a. Openr: An open source framework for advanced reasoning with large language models. *arXiv preprint arXiv:2410.09671*.

Qixin Wang, Dawei Wang, Kun Chen, Yaowei Hu, Puneet Girdhar, Ruoteng Wang, Aadesh Gupta, Chaitanya Devella, Wenlai Guo, Shangwen Huang, and 1 others. 2025b. Adaptjobrec: Enhancing conversational career recommendation through an llm-powered agentic system. *arXiv preprint arXiv:2508.13423*.

Shijie Wang, Wenqi Fan, Yue Feng, Shanru Lin, Xinyu Ma, Shuaiqiang Wang, and Dawei Yin. 2025c. Knowledge graph retrieval-augmented generation for llm-based recommendation. *arXiv preprint arXiv:2501.02226*.

Shuhe Wang, Shengyu Zhang, Jie Zhang, Runyi Hu, Xiaoya Li, Tianwei Zhang, Jiwei Li, Fei Wu, Guoyin Wang, and Eduard Hovy. 2024b. Reinforcement learning enhanced llms: A survey. *arXiv preprint arXiv:2412.10400*.

Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. 2020. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020*, pages 1082–1092.

Ziyan Wang, Yingpeng Du, Zhu Sun, Haoyan Chua, Kaidong Feng, Wenya Wang, and Jie Zhang. 2025d. Re2llm: reflective reinforcement large language model for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12827–12835.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, and 1 others. 2025. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning. *arXiv preprint arXiv:2505.16421*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*.

Dayu Yang, Fumian Chen, and Hui Fang. 2024. Behavior alignment: A new perspective of evaluating llm-based conversational recommendation systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2286–2290.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

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

Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and 1 others. 2024. Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6889–6907.

Lixi Zhu, Xiaowen Huang, and Jitao Sang. 2025a. A llm-based controllable, scalable, human-involved user simulator framework for conversational recommender systems. In *Proceedings of the ACM on Web Conference 2025*, pages 4653–4661.

Yaochen Zhu, Chao Wan, Harald Steck, Dawen Liang, Yesu Feng, Nathan Kallus, and Jundong Li. 2025b. Collaborative retrieval for large language model-based conversational recommender systems. In *Proceedings of the ACM on Web Conference 2025*, pages 3323–3334.

Xingchen Zou, Yuhao Yang, Zheng Chen, Xixuan Hao, Yiqi Chen, Chao Huang, and Yuxuan Liang. 2025. Traffic-r1: Reinforced llms bring human-like reasoning to traffic signal control systems. *arXiv preprint arXiv:2508.02344*.

APPENDIX

A Dataset

In this study, we conduct experiments based on RecBench+ (Huang et al., 2025a), which provides a clear difficulty hierarchy of user queries and covers diverse scenarios requiring complex reasoning, such as multi-hop reasoning and reflection. The queries are categorized into two main types: **Condition-based Query**, which encompasses hard constraints like directors or actors, and **User Profile-based Query**, which includes softer criteria such as user preferences or mood. Examples of these queries are presented in Table 5.

B Baselines

As traditional recommendation systems like GRU4Rec (Hidasi et al., 2015) and SASRec (Kang and McAuley, 2018) cannot process natural language queries, in our experiments, we compare our method against three categories of approaches that can process natural language queries. Below is detailed information about these methods:

- **LLM Backbone:** This category employs pretrained LLM backbones directly for recommendation tasks.
  - **Qwen-2.5-3B-Instruct:** Qwen-2.5-3B-Instruct is a 3 billion parameter instruction-tuned LLM from the Qwen series, designed for enhanced performance in coding, mathematics, and instruction following tasks.
  - **Llama-3.2-3B-Instruct:** The Llama 3.2 collection of multilingual large language models (LLMs) is a collection of pretrained and instruction-tuned generative models in 1B and 3B sizes. The Llama 3.2 instruction-tuned text only models are optimized for multilingual dialogue use cases, including agentic retrieval and summarization tasks.
  - **DeepSeek-R1-Distill-Qwen-7B**: DeepSeek-R1-Distill-Qwen-7B is derived from Qwen-2.5 series, which are finetuned with 800k samples curated with DeepSeek-R1.
  - **GPT-4o:** GPT-4o is a multimodal LLM developed by OpenAI, capable of processing and generating text, images, and audio.
  - **DeepSeek-R1 (Guo et al., 2025):** DeepSeek-R1 is an open-source reasoning model developed by DeepSeek, trained using RL techniques to achieve state-of-the-art performance in reasoning tasks.
- **LLM-based Conversational Recommender Systems (CRS):** These methods are LLM-based CRS designed for recommendation tasks.
  - **TallRec (Bao et al., 2023):** Utilizing LoRA technique to fine-tune LLMs on recommendation dataset.
  - **InteRecAgent (Huang et al., 2025c):** InteRecAgent is a framework that combines LLMs with traditional recommender systems, enabling interactive and conversational recommendations by leveraging the strengths of both paradigms. It employs LLMs as the "brain" and uses recommender models as tools, incorporating components like memory, task planning, and reflection to transform traditional systems into interactive ones with natural language interfaces.
  - **CRAG (Zhu et al., 2025b):** CRAG is a conversational recommender system that combines LLMs with collaborative filtering techniques, providing context-aware and personalized recommendations.
- **RFT-based Methods:** RLVR-based Methods include popular RL algorithms with verifiable rewards, focusing on improving LLM reasoning capabilities without extensive human annotation.
  - **GRPO (Shao et al., 2024):** GRPO is an algorithm for training LLMs with RL. For each question, GRPO randomly sample multiple answers, and the advantage of an answer is defined as the normalized reward, thereby getting rid of the value estimation network.
  - **REINFORCE++ (Hu, 2025):** REINFORCE++ is an enhanced variant of the classical REINFORCE (Williams, 1992) algorithm, incorporating optimization techniques from Proximal Policy Optimization (PPO) (Schulman et al., 2017) while eliminating the need for a critic network.
  - **RLOO (Ahmadian et al., 2024):** RLOO (REINFORCE Leave One-Out) is a RL method that leaves out one sample for evaluation, ensuring robust alignment with human preferences.

C Prompt

We use below prompt template to guide the LLM in generating responses.

Table 5: Examples of RecBench+

Task	Query	Required Capability
Explicit Condition Query	I'm really interested in classic films and would love to watch something that showcases **Charlie Chaplin**'s legendary comedic talent. Additionally, I've heard that **Roland Totheroh's cinematography** adds an exceptional visual quality to movies. If you could point me in the direction of films that include both of these elements, I'd greatly appreciate it!	**Direct Reasoning:** Identifies specific attributes (e.g., director, cinematographer) and matches them directly.
Implicit Condition Query	I recently **watched Clockers (1995)** and **Bamboozled (2000)**, and I was really impressed by the direction in both films. I'm eager to explore more works from the director, as I found their storytelling style and vision very engaging. If you could suggest other films **associated with this director**, that would be fantastic.	**Multi-hop Reasoning:** Infers the attribute from given information and then generate corresponding recommendations.
Misinformed Condition Query	I recently watched **Lorenzo's Oil** and was really impressed by the cinematography done by Mac Ahlberg. I'm interested in finding more films that showcase his cinematographic style. I also remember seeing his work in **Beyond Rangoon**, so if there are any other movies he contributed to, I'd love to check them out!	**Reflection:** Detects and corrects misinformation (e.g., Mac Ahlberg did not work on these films) before generating recommendations.
Interest-based Query	I'm fond of **romantic and dramatic** films from the **golden age of Hollywood** like 'Roman Holiday' and 'My Fair Lady'. Are there any other dramatic romances from that period you would recommend?	**Contextual Reasoning:** Leverages user interest context to suggest similar content.
Demopraphics-based Query	I'm a **psychology professor** and I'm looking for movies that delve into **human emotions and relationships**. Have you got any?	**Domain-specific Reasoning:** Applies demographic details (e.g., occupation, age, gender) to recommend relevant content.

Prompt Template

Given the user's query, select one {movie\book} that best matches the query from candidates. You should think step-by-step and explain why you choose this {movie\book} and concisely why you didn't choose the others. The final answer should be in \\boxed{}.
Query: {query}
Candidates: {candidate list}

D Implementation Details

D.1 RQ1

Experiments were conducted on 2 H20 GPUs (96GB). The prompt template for the rollout process is detailed in **Appendix C**. For our method and all RFT-trained baselines, we use a learning rate of 5e-6, a group size of 5, and set the maximum response length to 768. Training is conducted for up to 15 epochs with early stopping (patience = 1). In our method, the hyperparameter $w_{penalty}$ in RAEE is set to 0.3. For the dual-graph-enhanced reward module, the weights w_1 and w_2 are set to equal values of 0.01 on the movie and book dataset. The difficulty threshold τ used in the online curriculum scheduler is set to 0.1. The implementation utilized the PyTorch 2.6.0, Verl 0.3.1, VLLM 0.8.5, and Ray 2.46.0. The training-related hyperparameters are specified as follows: a batch size of 256 for policy updates, a KL loss coefficient of 0.01, a rollout temperature of 1.0, and a clipping ratio ε of 0.2. For the movie and book domains, we sample 10,000 queries each as the training set and 12,000 queries each as the test set. Training is conducted separately on these two domains.

D.2 RQ2

We select Condition-based Queries that are not present in the training set. For each query, the candidates are constructed as follows: one positive item, three hard negatives (items that satisfy the conditions in the query but are not present in the user's interaction history), and sixteen simple negatives (randomly sampled items).

For the "without history" setting, we use the prompt from Appendix C to allow the trained

ReRec model (with Qwen-2.5-3B-Instruct as the backbone) to make predictions. In the "with history" setting, we build upon the previous prompt and introduce the user's {watching/reading} history before the candidates. This history consists of 10 randomly sampled items from the user interaction history.

D.3 RQ3

Cross-Domain Setting: For the cross-domain evaluation, we directly apply the model trained on the Movie domain to make predictions in the Book domain, and vice versa. As a baseline, we include the original base model without any task-specific training.

Cross-Task Setting: To assess generalization across tasks, we apply ReRec —trained on our task—to the sequential recommendation task. In this setup, the model is given a user's interaction history and asked to predict the next item. We compare performance against the base model and standard recommendation baselines, including GRU4Rec (Hidasi et al., 2015) and SASRec (Kang and McAuley, 2018). The maximum length of the interaction history is set to 10, and all models are required to select the most likely item from a pool of 20 candidates.

D.4 RQ4

Supervised Fine-Tuning (SFT) Details: We fine-tuned the Qwen-2.5-3B-Instruct model using full parameter training, with the input provided by Prompt C and the corresponding target items as the output. Training was conducted on an H20 GPU (96GB memory) using a learning rate of 1e-5. We applied early stopping with a patience of 1 to prevent overfitting.

Evaluation Details: We evaluated the Qwen-2.5-3B-Instruct, the fine-tuned SFT model, and ReRec across a series of benchmarks. All models were tested under identical inference settings to ensure fair comparison. Below is the information of the used benchmarks:

- **DROP:** The DROP [1] (Discrete Reasoning Over Paragraphs) benchmark is designed to evaluate the reading comprehension and reasoning capabilities of AI models. It includes a variety of tasks that require models to read passages and answer questions based on the content. We evaluate models with zero-shot setting and use accuracy as evaluation metric.
- **IFEval:** IFEval [2] is a benchmark for evaluating instruction-following language models, focusing on their ability to understand and respond to various prompts. It includes a diverse set of tasks and metrics to assess model performance comprehensively. We evaluate models with zero-shot setting and use accuracy as evaluation metric.
- **ARC:** ARC (AI2 Reasoning Challenge) [3] benchmark is designed to evaluate the reasoning capabilities of AI models through multiple-choice questions derived from science exams. It includes two subsets: ARC-Easy and ARC-Challenge, which vary in difficulty. We evaluate models with zero-shot setting and use accuracy as evaluation metric.
- **GPQA:** GPQA [4] is a multiple-choice, Q&A dataset of very hard questions written and validated by experts in biology, physics, and chemistry. When attempting questions out of their own domain (e.g., a physicist answers a chemistry question), these experts get only 34% accuracy, despite spending >30 minutes with full access to Google. We evaluate models with a 5-shot setting and use pass@1 as an evaluation metric.

E Case Study

To intuitively demonstrate ReRec 's reasoning capabilities on interactive recommendation, we present a representative case study below. The user query seeks films shot by the same cinematographer as Pan's Labyrinth (2006) and Four Rooms (1995), emphasizing their distinctive visual storytelling. ReRec delivers a precise recommendation through the following step-by-step reasoning process, showcasing its ability to handle complex, context-driven queries effectively.

- **Multi-hop Reasoning:** ReRec correctly identifies **Guillermo Navarro** as the cinematographer for *Pan's Labyrinth* and links his work to *The Long Kiss Goodnight (1996)* and *Hellboy (2004)* from the candidate list.
- **Contextual Evaluation:** Beyond merely matching cinematographers, ReRec evaluates stylistic alignment, dismissing *Hellboy (2004)*—another Navarro film—due to its divergence from the visual style of the user's preferred films. This

[1] https://modelscope.cn/datasets/AI-ModelScope/DROP/summary

[2] https://modelscope.cn/datasets/opencompass/ifeval/summary

[3] https://modelscope.cn/datasets/modelscope/ai_arc/summary

[4] https://huggingface.co/datasets/Idavidrein/gpqa

Case for ReRec Reasoning

Query: I recently watched Pan's Labyrinth (El Laberinto del fauno) (2006) and Four Rooms (1995) and was really captivated by the cinematography. The visual storytelling in both films left a strong impression on me, and I'm eager to explore more movies that share this distinctive visual style. If you could suggest films that were shot by the same cinematographer, that would be great! **Candidates:** Stuart Little 2 (2002);Fabulous Baker Boys, The (1989);Hellboy (2004);Last Supper, The (1995);Twin Dragons (Shuang long hui) (1992);Go (1999);Escape from New York (1981);Hanging Garden, The (1997);They Made Me a Criminal (1939);King Kong (1933);Terms of Endearment (1983);Right Stuff, The (1983);**The Long Kiss Goodnight (1996)**;They Bite (1996);Home Alone 2: Lost in New York (1992);Tora! Tora! Tora! (1970);Pan's Labyrinth (El Laberinto del fauno) (2006);Devil Girl From Mars (1954);Van, The (1996);Iron Eagle IV (1995)

ReRec: I understand the user's preference for visually striking films and the specific cinematography style mentioned. To narrow down the options, I'll first identify which cinematographers are known for utilizing striking visual storytelling and cinematography. One strong candidate here could be Guillermo Navarro, who has directed both films that the user liked. He is known for his visually distinctive works.
Given the candidates:

- Guillermo Navarro is associated with films like **The Long Kiss Goodnight (1996)**.
- **Hellboy (2004)** also have Guillermo Navarro as the director, but its style is not very similar to the two movies mentioned by the user, so it may not be the best choice.

The Long Kiss Goodnight (1996) is the movie I believe best portrays the distinctive visual style mentioned by the user and credits the same cinematography team as both films the user was captivated by. Other options were not as relevant to the context and criteria specified.

reflects a nuanced understanding of user intent and context.

F Future Work

While ReRec establishes a robust framework for enhancing reasoning capabilities in LLM-based recommendation assistants, several promising directions remain open for future exploration.

Multi-turn Conversational Recommendation with Reasoning. One compelling future direction is extending ReRec to multi-turn conversational settings, where user intents evolve dynamically through dialogue. In such scenarios, the task becomes significantly more complex: the model must track contextual dependencies, resolve coreference, and integrate intermediate reasoning steps across utterances. Designing effective reward models under this setting is non-trivial. Unlike single-turn queries, multi-turn conversations require the model to maintain dialogue coherence and long-term goal alignment. Future work could explore hierarchical reward structures that combine utterance-level relevance with cumulative dialogue success.

Reasoning with Tools for Recommendation. Another important direction is augmenting ReRec with external tools to enable tool-augmented reasoning. While LLMs possess broad linguistic and general reasoning capabilities, real-world recommendation often requires interfacing with external systems—such as retrieval engines, structured databases, or collaborative filtering models—to access up-to-date or user-specific information. To this end, future research could explore LLM-agent architectures for recommendation, where the model acts as a planner that issues sub-tasks to external tools and integrates the responses via step-by-step reasoning.