Think before Recommendation: Autonomous Reasoning-enhanced Recommender

Xiaoyu Kong[1] **Junguang Jiang**[1] **Bin Liu**[1] **Ziru Xu**[1]
Han Zhu[1] **Jian Xu**[1] **Bo Zheng**[1] **Jiancan Wu**[3,4*] **Xiang Wang**[2*]
[1]Taobao & Tmall Group of Alibaba, China
[2]National University of Singapore
[3]Institute of Dataspace, Hefei Comprehensive National Science Center
[4]Shanghai Key Laboratory of Data Science
linlin.kxy@alibaba-inc.com
wujcan@gmail.com
xiangwang1223@gmail.com

Abstract

The core task of recommender systems is to learn user preferences from historical user-item interactions. With the rapid development of large language models (LLMs), recent research has explored leveraging the reasoning capabilities of LLMs to enhance rating prediction tasks. However, existing distillation-based methods suffer from limitations such as the teacher model's insufficient recommendation capability, costly and static supervision, and superficial transfer of reasoning ability. To address these issues, this paper proposes RecZero, a reinforcement learning (RL)-based recommendation paradigm that abandons the traditional multi-model and multi-stage distillation approach. Instead, RecZero trains a single LLM through pure RL to autonomously develop reasoning capabilities for rating prediction. RecZero consists of two key components: (1) "Think-before-Recommendation" prompt construction, which employs a structured reasoning template to guide the model in step-wise analysis of user interests, item features, and user-item compatibility; and (2) rule-based reward modeling, which adopts group relative policy optimization (GRPO) to compute rewards for reasoning trajectories and optimize the LLM. Additionally, the paper explores a hybrid paradigm, RecOne, which combines supervised fine-tuning with RL, initializing the model with cold-start reasoning samples and further optimizing it with RL. Experimental results demonstrate that RecZero and RecOne significantly outperform existing baseline methods on multiple benchmark datasets, validating the superiority of the RL paradigm in achieving autonomous reasoning-enhanced recommender systems. Our codes are available at https://github.com/AkaliKong/RecZero.

1 Introduction

Recommender models aim to learn user preferences on items from historical user-item interactions (*e.g.,* ratings, clicks, purchases) [1–7]. Motivated by the rapid progress of large language models (LLMs) [8–11], recent research has explored adapting LLMs as recommenders, leveraging their strengths in world knowledge, semantic understanding, and reasoning. A promising direction is to leverage LLMs' reasoning capabilities [12–16] to enhance rating prediction [17–19] — a core recommendation task that explicitly models user preferences by predicting their ratings on items.

Upon scrutinizing prior studies on such reasoning-enhanced rating predictors [20–25], we can summarize a common pipeline: (1) Given a user's rating on a target item, a teacher model — often a

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

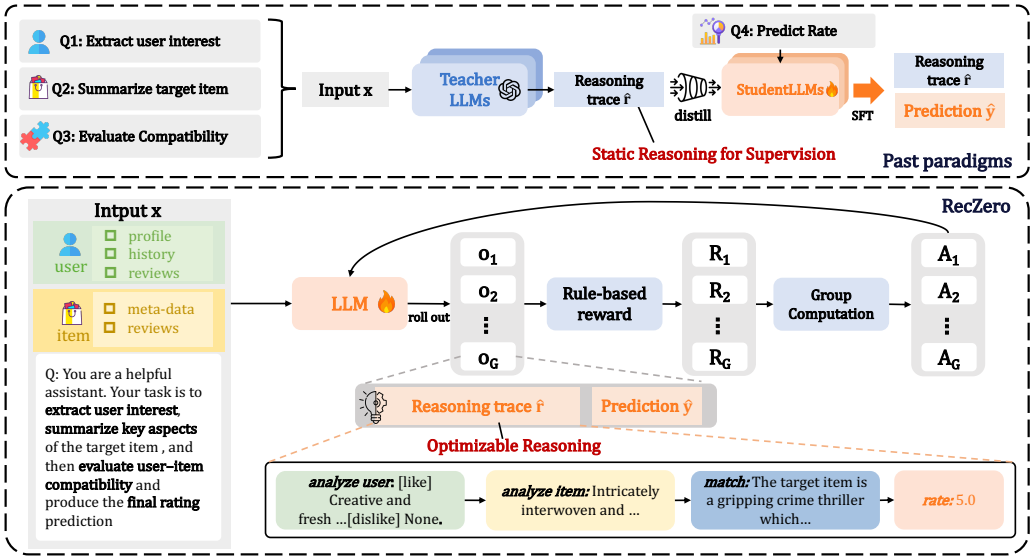


Figure 1: Comparison between RecZero and Conventional Paradigms. Conventional paradigms train multiple student LLMs using diverse query-based datasets with reasoning traces, while RecZero uses pure RL to train a single LLM for the entire workflow. Unlike conventional methods reliant on Teacher models, RecZero leverages Reward Signals to jointly optimize reasoning traces and predictions, improving efficiency and performance.

strong general-purpose LLM such as ChatGPT [8] — first extracts the user's preferences over item attributes from historical interactions, then generates possible intermediate reasoning steps to interpret the observed rating; (2) Each reasoning trace, paired with the corresponding user rating, forms an instruction sample; (3) A smaller model is then supervised fine-tuned on these reasoning-enhanced instruction samples to distill the teacher model's extraction and reasoning capabilities.

Despite recent progress, we identify several limitations inherent in this distillation paradigm, as evidenced in Figure 1:

- **Limited Recommendation Capability of the Teacher.** General-purpose teacher models lack domain-specific knowledge for recommendation, producing reasoning traces that are misaligned with the rate prediction objective. Consequently, these suboptimal reasoning processes hinder the student's performance.
- **Costly and Static Supervision.** Generating high-quality reasoning data at scale, whether via human annotation [26] or LLM API calls [27, 28], is both time- and resource-intensive. Moreover, the student model is restricted to passively consuming static teacher outputs, with no chance to actively refine or optimize the reasoning process.
- **Superficial Transfer of Reasoning Ability.** While the student model fine-tuned on teacher-generated data can reproduce reasoning traces, it often imitates surface-level patterns, rather than acquiring genuine reasoning skills [29]. This results in overfitting to the distribution of teacher instructions, limiting generalization to unseen tasks [29, 30].

To address the limitations of existing distillation-based methods, we draw inspiration from the recent success of reinforcement learning (RL) in LLM post-training [31–33], particularly DeepSeek-R1-Zero [32], and propose a simple yet effective RL paradigm for LLM-based recommendation, dubbed **RecZero**. Discarding the distillation paradigm that requires different models and disjoint stages (*i.e.,* information extraction and reasoning process generation of the teacher model, supervised fine-tuning of the student model), RecZero trains a single LLM via pure RL to develop autonomous reasoning capabilities for rating prediction. It is composed of two key components:

- **"Think-before-Recommendation" Prompt Construction.** Beyond the standard prompt containing a user's history and a target item, we introduce a structured reasoning template with special tokens to elicit step-wise analysis: "`<analyze user>` ... `</analyze user>`" prompts the model

to extract user interest from her/his historical interactions; "<analyze item> ... </analyze item>" summarizes key aspects of the target item; "<match> ... </match>" evaluates user–item compatibility based on the analyzed information; and consequently "<rate> ... </rate>" produces the final rating prediction. It decomposes the rating prediction task into discrete steps and employs chain-of-thought reasoning.

- **Rule-based Reward Modeling.** We adopt group relative policy optimization (GRPO) [31] to optimize the LLM with rule-based rewards. Given a prompt, the model samples multiple reasoning trajectories. For each trajectory, the reward is computed as the difference between the ground-truth rating and the predicted rating in the `rate` step. For the group of trajectories, we derive the relative advantage over the average reward as the signal to optimize the LLM.

As a result, unlike distillation-based methods that risk overfitting to surface-level reasoning traces, RecZero enables the LLM to acquire reasoning ability through interaction and optimization. The unified reasoning steps — <analyze user>, <analyze item>, <match>, and <rate> — are jointly trained to reflect and refine toward better recommendation performance, guided by recommendation-specific objectives. Beyond the pure RL approach of RecZero, we also explore a hybrid paradigm inspired by RL with cold start [32], termed **RecOne**. RecOne first constructs a small set of cold-start reasoning samples tailored for recommendation tasks, which are used to initialize the LLM via supervised fine-tuning. This warm-start model is then further optimized using RL to enhance its reasoning capabilities. By combining data-efficient initialization with task-specific RL, RecOne aims to achieve faster convergence and stronger performance in recommendation reasoning. We assess the effectiveness of RecZero and RecOne through extensive experiments on four benchmark datasets (*e.g.,* Amazon-book, Amazon-music [34], Yelp [35], IMDb [36]), showcasing the RL paradigm's superiority over distillation methods (*e.g.,* Rec-SAVER [20], EXP3RT [22], Reason4Rec [21]).

2 Preliminary

2.1 Reasoning-enhanced Rating Prediction

The primary goal of rating prediction task is to predict the user's ratings for items that align with user preferences. Let $\mathcal{U}$ denote a set of users, $\mathcal{I}$ a set of items, and $\mathcal{Y}$ the set of possible ratings. Formally, consider a user $u \in \mathcal{U}$ with a historical interaction sequence represented as $\mathcal{H}_u =< h_{u,1}, h_{u,2}, ..., h_{u,t} >$, where $h_{u,i} = (\mathcal{M}_i, y_{u,i}, d_{u,i})$ is a triplet that includes the meta-information $\mathcal{M}_i$ of the item i, the user's rating $y_{u,i} \in \mathcal{Y}$ of the item, and the user's review $d_{u,i}$. The prediction rule is defined as follows:

$$\hat{y}_{u,i} = \underset{k \in \mathcal{Y}}{\text{argmax}} \mathbb{P}_\theta(y_{u,i} = k | \mathcal{H}_u, \mathcal{M}_i) \tag{1}$$

where $\hat{y}_{u,i}$ denotes the predicted rating for user u on item i, θ represents the model parameters.

In the context of reasoning-enhanced rating prediction, LLM parametered by θ receives the interaction history $\mathcal{H}_u$ of user u and the meta-information $\mathcal{M}_i$ of the item i as input $x_{u,i}$. Subsequently, the LLM performs an explicit reasoning process and makes the final user rating prediction, expressed as:

$$\hat{r}_{u,i}, \hat{y}_{u,i} = LLM(x_{u,i}) = LLM(\mathcal{H}_u, \mathcal{M}_i) \tag{2}$$

where $\hat{r}_{u,i}$ is introduced as the reasoning trace and $\hat{y}_{u,i}$ represents the predicted rating. In our framework, the reasoning process is decomposed into analyzing user preferences and extracting the characteristics of the target item, followed by a matching analysis.

3 Methodology

To address the major challenges inherent in existing reasoning-enhanced rating prediction methods, we propose RecZero. This innovative approach leverages pure reinforcement learning to encourage the model to jointly optimize the four critical processes of user analysis, item analysis, matching, and rating under a unified recommendation-specific objective. To further enhance the model's performance in recommendation tasks, we introduce RecOne, building upon RecZero. RecOne employs a specialized sampling method to obtain high-quality reasoning traces, which are used to perform SFT for cold-starting the initial model. Subsequently, we conduct task-specific RL to achieve faster convergence and more robust recommendation performance, as illustrated in Figure 2.

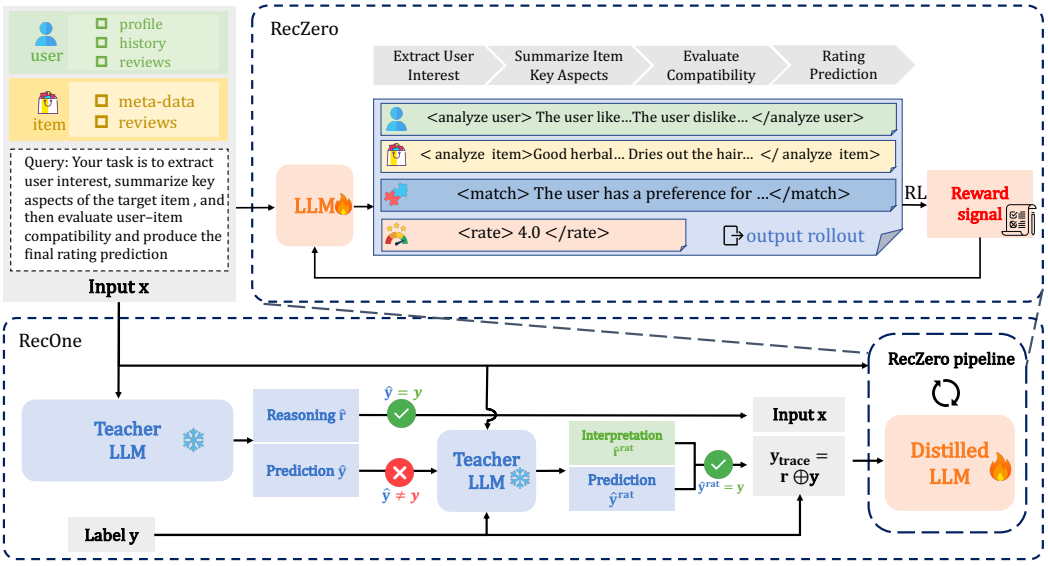


Figure 2: The RecZero and RecOne frameworks utilize a unified target to optimize multi-step reasoning within a coupled structure, achieving reasoning-enhanced predictions specifically designed for recommendation scenarios.

3.1 RecZero: Pure RL Paradigm for Reasoning-enhanced Recommendation

3.1.1 "Think-before-Recommendation" Prompt Construction

Previous studies [21, 22] have demonstrated that in reasoning-enhanced recommendation scenarios, requiring LLMs to first explicitly extract user preferences and item features, and then deliberate on the match between users and items, can significantly enhance the accuracy of LLMs in predicting users' actual ratings. High-quality rating predictions often rely on clear and high-quality extraction processes, whereas user preferences and item features typically lack explicit ground truth. To address this, we mandate that LLMs strictly output user preferences and item feature extractions according to predefined rules, and employ a rule-based reward mechanism to guide LLMs in self-optimization without the need for labeled data, thereby replacing the previous approach of using frozen LLMs with few-shot learning for these tasks. We have designed a system prompt to guide the LLM in generating reasoning trajectories beneficial to the rating prediction task. Specifically, we divide the reasoning process into three parts: user interest extraction, item aspects summarization, and compatibility evaluation. For a given user history and item metadata, we mandate that the LLM constrains the outputs of different steps within the specified token boundaries, as shown in Appendix.

3.1.2 Rule-based Reward Modeling

Rule-based approaches [31] can provide stable training rewards for RL. To address the issue of suboptimal performance caused by inconsistent training objectives in previously decoupled modules, we integrate user preference summarization, item feature distillation, and match deliberation within a unified framework and optimize them through a unified reward rule. For the format reward, we first define the correct format as follows:

1. The model's thinking process should be encapsulated within the `<analyze user>` $\cdots$ `</analyze user>`, `<analyze item>` $\cdots$ `</analyze item>`, and `<match>` $\cdots$ `</match>` tags, respectively.
2. The generated output must be within the `<rate>` $\cdots$ `</rate>` tags and free of any unreadable content.

Based on the above format requirements, the format reward is defined as follows:

$$R_{format} = \begin{cases} 0.5 & \text{if the format is correct} \\ -0.5 & \text{if the format is incorrect} \end{cases}, \tag{3}$$

In our training process, adhering to the continuous rating paradigm established in prior work [21, 22], we employ Rule-based Reward Modeling to encourage the model to approximate actual user ratings as closely as possible, rather than relying solely on correctness rewards. This objective can be formally expressed as:

$$R_{answer} = 1 - \frac{|y - \hat{y}|}{\text{max\_error}}, \tag{4}$$

where max_error is the maximum allowable error (which can be set based on the task, *e.g.,* when the rating range is from 1 to 5, max_error = 4). The final total reward can be formally expressed as $R = R_{format} + R_{answer}$. Unlike previous LLM-based paradigms, we do not design a separate task that requires the LLM to output an integer score that is then converted into a decimal rating via logit-weighted decoding. Compared with the SFT paradigm that imitates integer target labels, reinforcement learning offers a clear advantage: by crafting an appropriate reward function, we encourage the LLM to predict decimal ratings directly in its response, thereby earning a higher MAE reward and removing the need for the logit-weighted step altogether.

3.1.3 Group Relative Policy Optimization

We use Group Relative Policy Optimization (GRPO) [31] for RL policy optimization. In this approach, for a given input x[1], the method generates a set of outputs $\{y_1, y_2, \ldots, y_G\}$ using the existing policy $\pi_{\theta_{\text{old}}}$. The policy model is then refined by maximizing the following objective function:

$$\begin{aligned} J_{\text{GRPO}}(\theta) = & \mathbb{E}_{x\sim D,\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \left\{ \min\left(\frac{\pi_\theta(y_{i,t}|x,y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x,y_{i,<t})} \hat{A}_{i,t}, \right.\right.\right. \\ & \left.\left.\left. \text{clip}\left(\frac{\pi_\theta(y_{i,t}|x,y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x,y_{i,<t})}, 1-\epsilon, 1+\epsilon \right) \hat{A}_{i,t} \right) - \beta \mathbb{D}_{\text{KL}}\left[\pi_\theta \| \pi_{\text{ref}}\right] \right\} \right], \end{aligned} \tag{5}$$

where β is a parameter that adjusts the trade-off between the task-specific loss and the KL-divergence. The advantage term $\hat{A}_i$ is derived from the rewards of the group of responses $\{R_1, R_2, \ldots, R_G\}$ and is calculated as:

$$\hat{A}_i = \frac{R_i - \text{mean}(\{R_1, R_2, \ldots, R_G\})}{\text{std}(\{R_1, R_2, \ldots, R_G\})}. \tag{6}$$

This formulation ensures that the policy optimization process is guided by the relative performance of the generated outputs within each group, promoting more stable and efficient learning.

3.2 RecOne: Distillation and RL Paradigm for Reasoning-enhanced Recommendation

While pure reinforcement learning approaches like RecZero show promise, we recognize opportunities for further enhancements. Specifically, the discrepancy between pretrained LLMs' training corpus and recommendation-specific data creates a domain gap that the RL paradigm must gradually bridge. Previous studies [37, 38] have demonstrated that LLMs can enhance their capabilities through self-generated reasoning processes. In this light, we propose a hybrid paradigm, RecOne, which first establishes a solid foundation through cold-start supervised fine-tuning on high-quality reasoning trajectories, then leverages the RecZero framework to further refine its reasoning capabilities.

3.2.1 Cold-Start Supervised Fine-Tuning

The first phase of RecOne addresses the foundational reasoning capabilities necessary for effective recommendations. We first perform a warm-up using recommendation data with long reasoning trajectories. Specifically, we sample M data samples to construct a subset D_{sub} from the complete dataset D, which is utilized in RecZero. For each data pair $(x, y) \in D_{sub}$, we leverage a teacher LLM (*e.g.,* DeepSeek-R1) with superior reasoning capabilities to generate a reasoning trajectory $(\hat{r}, \hat{y})$ given the input x, following Equation (2). When the predicted rating $\hat{y}$ aligns with the ground truth y, we consider the generated reasoning $\hat{r}$ to be high-quality; While for cases where $\hat{y}$ does not match y, we feed both x and y to the teacher model, requiring it to generate a rationalized trajectory $\hat{r}^{rat}$ that

[1]For notational clarity, we omit the user and item indices, while retaining other subscripts in the subsequent parts.

aligns with the correct rating. This process creates a reasoning trajectory dataset that consists of two complementary subsets:

$$D_{\text{trace}} = D_{\text{align}} \cup D_{\text{misalign}}, \tag{7}$$

$$D_{\text{align}} = \{(x, \hat{r} \oplus y) \mid \hat{y} = y\}, \tag{8}$$

$$D_{\text{misalign}} = \{(x, \hat{r}^{\text{rat}} \oplus y) \mid \hat{y} \neq y \wedge \hat{y}^{\text{rat}} = y\}. \tag{9}$$

Here, D_{align} represents examples where the teacher LLM's reasoning naturally leads to the correct rating prediction, while D_{misalign} captures cases requiring corrective reasoning to reach the proper conclusion. The training objective is formulated as an autoregressive model optimization problem:

$$\max_{\theta} \sum_{(x, y^{\text{trace}}) \in D_{\text{trace}}} \sum_{t=1}^{|y^{\text{trace}}|} \log P_\theta(y_t^{\text{trace}} | x, y_{<t}^{\text{trace}}), \tag{10}$$

where y^{trace} represents the concatenation of reasoning trajectory and ground-truth rating as in Equations (8) and (9), θ denotes the LLM's model parameters, y_t represents the t-th token in the output sequence, and $y_{<t}$ includes all preceding tokens in the sequence.

4 Experiments

In this section, we first demonstrate the performance improvement of the RecZero and RecOne framework on the rating prediction task. We requested the dataset from Reason4Rec [21] and conducted our experiments based on it. We conduct extensive experiments on real-world datasets, including Amazon-book, Amazon-music, and Yelp, to evaluate the effectiveness of the proposed RecZero and RecOne framework. Our analysis includes a detailed comparison of RecOne with existing baseline models, which cover CF-based models [39], Review-based models [40, 17, 18] and LLM-based recommendation models [20–22, 25].

We employed Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as evaluation metrics to reflect the model's accuracy in rating prediction. More details about these baseline models, datasets, and metrics can be found in Appendix. Additionally, we conducted comprehensive ablation studies to identify the key components that enhance the performance of RecOne, with a particular focus on the roles of the "thinking process" and the "format-guided model." Besides accuracy studies, we also compare the training and inference cost of RecZero against previous LLM-based paradigms. In short, we aim to address the following research questions:

- **RQ1:** How does RecZero and RecOne perform in comparison to other baseline methods?
- **RQ2:** Does the initial capability of the model have an impact on the upper limit of RL performance?
- **RQ3:** What is the impact of the designed components on the recommendation performance of RecOne?
- **RQ4:** How efficient is RecZero in terms of training and inference cost compared with the prior reasoning-enhanced baseline?

4.1 Performance Comparison (RQ1)

We conduct a holistic evaluation, considering metrics of both MAE and RMSE across Book, Music and Yelp datasets to demonstrate the effectiveness of our framework. This section comprehensively compares RecZero and RecOne against CF-based, Review-based and LLM-based baselines.

From Table 1 we observe that RecOne ranks first on all six evaluation metrics across the three domains. Concretely, it lowers the best previous RMSE by 6.7%, 12.2% and 6.2% on Book, Music and Yelp respectively, while cutting the MAE by 16.8%, 29.9% and 7.5%.When it relies solely on RL inside the LLM, with neither cold-start pre-initialization nor guidance from a teacher model, RecZero still surpasses all baselines in terms of MAE on the three datasets.

This section delves into the profound impact of the RecOne cold-start model on subsequent training processes. On the book dataset, we systematically compared the training trajectories of the distilled model obtained under the RecOne framework with those of the model trained using the initial model following the RecZero paradigm for RL, as illustrated in Figure 3.

Table 1: The Results of RecZero and RecOne compared with Traditional models and LLMs-based methods.

Methods	Book		Music		Yelp	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
CF-based						
MF	0.8565	0.6277	0.8142	0.6188	1.0711	0.7980
Review-based						
DeepCoNN	0.8403	0.6211	0.8057	0.6034	1.0665	0.8312
NARRE	0.8435	0.6242	0.7881	0.5799	1.0785	0.8177
DAML	0.8371	0.6214	0.7848	0.5703	1.0405	0.7964
LLM-based						
Rec-SAVER	0.9356	0.6645	0.9262	0.6463	1.1282	0.8295
EXP3RT	0.9346	0.6042	0.8312	0.5548	1.1420	0.8236
Reason4Rec	0.8325	0.5937	0.7647	0.5352	0.9972	0.7473
Ours						
RecZero	0.8387	0.5253	0.7058	0.4271	1.0521	0.7429
RecOne	**0.7784**	**0.5017**	**0.6776**	**0.3816**	**0.9774**	**0.7012**

4.2 In-Depth Analysis of RecZero and RecOne (RQ2)

Notably, the initial model of RecZero exhibited an anomalous increase in MAE during the first 20 training steps, primarily attributed to the initial LLM focusing on acquiring relatively easier-to-obtain format rewards. In contrast, the distilled model of RecOne demonstrated higher format scores at the onset of training, enabling it to optimize the answer scores for the rating task more rapidly during RL training. Throughout the displayed 200 training steps, the RecOne model consistently outperformed RecZero, a result that unequivocally demonstrates the effectiveness of the cold-start strategy in enhancing the model's initial capabilities, thereby significantly elevating its performance ceiling for RL optimization. During the early stages of RL training, the MAE drops sharply, then decreases slowly with minor fluctuations, and finally stabilizes over a longer training period.

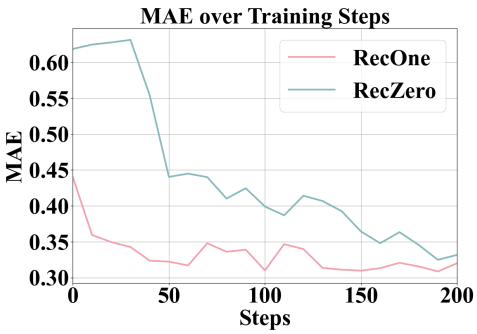
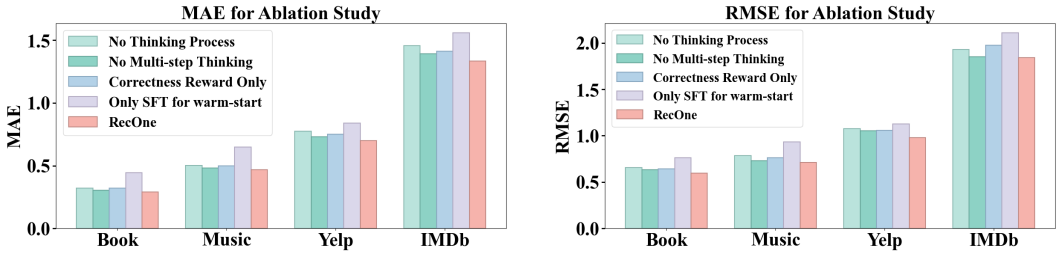


Figure 3: The performance of RecZero and RecOne over training steps.

4.3 Ablation Study (RQ3)

To validate the effectiveness of each component in the Rec-zero framework, we compare it with the following alternative approaches:

- **No Thinking Process:** We eliminate the required thinking process in the model output, requiring the model to directly predict the specific rating value based on the input.
- **No Multi-step Thinking:** We replace the multi-step thinking process `<analyze user>`, `<analyze item>` and `<match>` with a single-step `<think>` without imposing specific format requirements.
- **Correctness Reward Only:** We use a single correctness reward as the reward signal. Specifically, when the predicted score exactly matches the target, a correctness reward of 2 points is given; otherwise, it is assigned 0.
- **Only SFT for warm-start:** We discard the step in Rec-one that jointly optimizes reasoning and rating prediction through reinforcement learning, and instead only use the Teacher Model to generate trace data for cold-start scenarios.



(a) MAE (b) RMSE

Figure 4: 4a and 4b separately show the MAE and RMSE performance of RecOne and its four ablation variants on four datasets.

The experimental results are illustrated in Figure 4. Our RecOne model demonstrates significant superiority over alternative approaches in both MAE and RMSE metrics. Through experimental observations, we can discern that the **No Thinking Process** baseline underperforms the **No Multi-step Thinking** version in both metrics, which in turn is inferior to the complete RecOne model employing the three-step thinking process (Extract User Interest, Summarize Item Key Aspects, Evaluate Compatibility). This robustly validates the positive impact of our proposed strategy of explicitly guiding the model through multi-step thinking on rating prediction accuracy. Furthermore, the alternative version **Correctness Reward Only** exhibits a notable decline in both MAE and RMSE metrics compared to RecOne. This phenomenon can be attributed to the fact that the correctness-only reward mechanism encourages the model to generate integer ratings that exactly match user scores, rather than decimal values approximating the actual ratings. Such mechanism suppresses the model's tendency to produce predictions close to actual ratings, instead promoting aggressive predictions that completely align with target scores. Finally, the contrast model employing **Only SFT for warm-up** demonstrates the poorest performance among all baseline methods. This strongly substantiates the significance of our proposed approach of collaboratively optimizing the reasoning process and rating prediction task through RL, acquiring reasoning ability through interaction and optimization, rather than merely obtaining surface-level reasoning through SFT for enhancing the model's predictive accuracy.

4.4 Cost–Effectiveness and Practical Deployment (RQ4)

We follow exactly the same experimental protocol as in the previous sections and, on the Amazon-Music dataset, additionally evaluate two further variants:

- **RecZero(early–stop)** — training stops once it attains the same MAE as the external best SFT baseline;
- **RecOne(SFT)** — the RL phase is removed and the model is trained with SFT until convergence.

Table 2 reveal four main findings:

- **Larger performance gain per unit cost.** The pure-SFT variant RecOne(SFT) consumes 20 K labelled instances, $\sim$0.6 GPU-hours, and 597 inference tokens per request to reach an MAE of 0.6472. By contrast, RecZero(early–stop) already surpasses this score (MAE $=$ 0.5419) while using only 0.48 K labels (-97.6%), 0.4 GPU-hours (-33%), and significantly smaller serving budget (331 tokens). Extending RL to full convergence (RecZero) further pushes the error down to MAE/RMSE $=$ 0.4271/0.7058, achieving a 21.5% MAE reduction against the strongest pure-SFT baseline (Reason4Rec) with merely 12% of its labels—yielding more than a $5\times$ gain in *error reduction per 1K labels*.
- **Fast adaptation to cold-start and distribution drift.** Commercial recommenders face a constant influx of new items and shifting user interests. RecZero can be re-optimized with merely a few hundred fresh on-line interactions collected within minutes, whereas SFT pipelines must accumulate and curate a much larger batch before retraining.
- **The simplest pipeline among reasoning-enhanced methods.** RecZero keeps (1) a single model, (2) a single training stage, (3) no distillation teacher and (4) end-to-end optimization, resulting in markedly lower engineering overhead than multi-stage alternatives such as EXP3RT or Reason4Rec, which each juggle three models and at least two training phases.

Table 2: Training and inference cost of different reasoning–enhanced recommenders

Method	Paradigm	Samples	Models	Training Time	Avg.Inf.Tokens	Inf.Stages	Inf.Token
EXP3RT	SFT	20K	3	1.2h	187.63	3	562.89
Reason4Rec	SFT	20K	3	1h	175.49	2	350.98
RecZero(early-stop)	RL	0.48K	1	0.4h	331.64	1	331.64
RecOne(SFT)	SFT	20K	1	0.6h	597.32	1	597.32
RecZero	RL	2.4K	1	1.1h	310.52	1	310.52
RecOne	SFT+RL	2.6K	2	1.4h	412.79	1	412.79

- **RL complements rather than replaces SFT.** RecOne(full) shows that starting from an SFT warm-start and applying a short RL refinement can push the error down to MAE/RMSE = 0.3816/0.6776. In practice, one may first perform a quick SFT pass on large historical logs, then run a lightweight RecOne-style RL fine-tuning on the latest or strategically important traffic.

In summary, the experiments reveal four clear takeaways. First, pure SFT is outperformed by pure reinforcement learning: despite requiring 20 K labelled samples, longer training time and almost twice the inference cost, RecOne(SFT) still lags behind the early-stopped RL variant, RecZero(early-stop). Second, when allowed to converge fully, RecZero further cuts MAE by 21.5 % relative to the strongest SFT baseline, without demanding additional computational resources. Third, both RecZero and RecOne can be re-optimized using only a few hundred fresh interactions, making them well suited to sparse-data or drifting-distribution scenarios. Finally, both methods preserve a single-model, teacher-free pipeline, offering a markedly simpler and more cost-efficient solution than existing multi-stage reasoning-enhanced recommenders.

5 Related Work

5.1 LLM-based Explainable Recommendation

The remarkable capabilities demonstrated by LLMs in world knowledge understanding and contextual processing have inspired researchers to leverage LLMs for handling rich semantic information to provide recommendation suggestions [41–45]. DRDT [46] employs a prompt-based approach that requires LLMs to analyze user preferences from multiple dimensions, enhancing sequential recommendations through critic prompts for reflection. GOT4Rec [23] encourages LLMs to engage in multi-perspective thinking for recommendations using the graph of thought strategy. However, these methods primarily adopt few-shot approaches for recommendation tasks, constrained by the inherent limitations of LLMs in the recommendation domain.

Recent studies have explored fine-tuning LLMs to optimize their reasoning capabilities for recommendation. Rec-SAVER [20] utilizes larger-scale LLMs to generate rationalized intermediate reasoning, fine-tuning smaller models to enhance their recommendation reasoning abilities. EXP3RT and Reason4Rec [22, 21] decompose the reasoning process into multiple steps for independent training, distilling the reasoning capabilities of larger models into smaller ones, and separately optimizing rating tasks for better performance. While these efforts have significantly enhanced the accuracy of LLMs in performing reasoning-enhanced rating prediction tasks, challenges such as suboptimal reasoning trajectories and decoupled training objectives remain unresolved.

5.2 LLM Post-training

As the most widely-used post-training technique, SFT quickly adapts a pretrained LLM to downstream tasks, yet its impact on model generalization remains a matter of debate. Recent studies have investigated how SFT influences the generalization of foundation models. SFT Memorizes, RL Generalizes [30] shows that SFT tends to memorize training data and thus generalizes poorly to out-of-distribution scenarios, while RL strengthens a model's core abilities and enhances domain generalization. SFT or RL [29] systematically compares SFT and RL and reveals that SFT often leads to the imitation of "pseudo reasoning paths" generated by an expert model. These paths may look similar to the native reasoning produced by RL-trained models, yet they usually contain redundant, hesitant, or low-information steps and sometimes even incorrect reasoning.

RL [47, 48] operates through the interaction mechanism between agents and the environment, rewarding correct actions with the core objective of maximizing cumulative returns. In recent years,

researchers have introduced RL into the fine-tuning process of LLMs. Among these approaches, Reinforcement Learning with Human Feedback (RLHF) [49] typically employs the Proximal Policy Optimization (PPO) [50] algorithm to achieve alignment with human preferences. However, the PPO algorithm faces technical challenges related to excessive memory consumption. To simplify the application of RL in LLMs, researchers have proposed innovative methods such as Direct Preference Optimization (DPO) [51]. In the context of recommendation systems, the s-DPO method [52] combines DPO with the softmax loss function, enhancing LLMs' capability in mining hard negative samples for recommendation systems. Although these methods have improved efficiency, they still face challenges such as off-policy issues and performance ceilings that fall short of online methods. The recently proposed Group Relative Policy Optimization (GRPO) [31] further reduces memory requirements by estimating group scores, while replacing traditional reward models with rule-based reward mechanisms. Through this innovative paradigm, LLMs can significantly enhance their reasoning capabilities and domain-specific performance (e.g., in mathematics and programming) without requiring annotated data [32, 33]. Recent studies [53–55] have further explored the applications of RL in mathematics and general domains. Despite these advancements, the application of RL in improving rating prediction performance through reasoning capabilities remains unexplored.

6 Conclusion

In this work, we introduced RecZero, a novel RL paradigm for LLM-based recommendation systems, addressing key limitations of conventional distillation-based methods. By unifying reasoning and recommendation into a single LLM trained through structured prompts and rule-based rewards, RecZero eliminates the need for separate teacher-student models and enables continuous optimization of reasoning processes. Our experiments across multiple benchmarks demonstrate RecZero's superiority over existing approaches, highlighting its ability to develop autonomous reasoning capabilities aligned with recommendation objectives.

Additionally, we explored RecOne, a hybrid paradigm combining supervised fine-tuning with RL, which further elevates the performance ceiling of the model in RL. Both approaches showcase the potential of RL in advancing recommendation systems, paving the way for more adaptive and efficient models.

Acknowledgments and Disclosure of Funding

This research is supported by the Fundamental Research Funds for the Central Universities (WK2100250065) and the National Natural Science Foundation of China (62302321).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the contributions, including the proposed RecZero and RecOne paradigms, their advantages over distillation-based methods, and the experimental validation on benchmark datasets.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work performed in Appendix. By explicitly acknowledging these limitations, we provide a balanced view of our work and suggest directions for future research to address these challenges.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: we provide the full set of assumptions and a complete (and correct) proof in our methodology part 3. The assumptions are clearly stated, and the proofs are detailed to ensure correctness and reproducibility.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We fully disclose all the information needed in Appendix. This includes details about the datasets, experimental setups, hyperparameters, and evaluation metrics.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Due to internal company code requirements and user privacy concerns, we will not be providing open access to the data and code at the time of submission. However, we are actively working on this process and, after ensuring the security of the relevant code and data, we commit to open-sourcing them in the final version of the paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Justification: We provide open access to the data and code. And we explain our set in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We do it in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: For each experiment we proviede sufficient information in our experiments part Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our research fully conforms to the NeurIPS Code of Ethics. We have carefully reviewed the guidelines and ensured that all aspects of our research, from data collection and usage to experimental conduct and reporting, adhere to ethical standards. We maintain transparency in our methodology, respect for privacy, and fairness in our experimental design, ensuring that our work aligns with the ethical principles outlined by NeurIPS.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Our paper discusses the potential positive and negative societal impacts of our work in the "Broader Impacts" section. By acknowledging these impacts, we provide a balanced view of our work and suggest mitigation strategies to address potential negative outcomes.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We conduct experiments on recommendation tasks using publicly available datasets and build upon existing open-source methods. Therefore, the risk of misuse is minimal, and specific safeguards for data or model release are not necessary. Our focus remains on transparency and reproducibility, ensuring that our work can be used responsibly by the research community.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We fully adhere to these requirements by properly crediting the creators or original owners of all assets used in our paper. Each asset, including datasets, code, and models, is accompanied by a citation to the original source, and the specific versions used are stated. We explicitly mention the licenses and terms of use for each asset, ensuring compliance with their respective usage guidelines.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We commit to open-sourcing our code and datasets in the officially published version.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: We are not involved in these risks as we are only engaged in recommendation tasks. Our research does not include crowdsourcing experiments or research with human subjects, thus this question is not applicable to our paper.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We are not involved in these risks as we are only engaged in recommendation tasks. Our research does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLM is used only for formatting purposes.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

References

[1] J. Ben Schafer, Joseph A. Konstan, and John Riedl. Recommender systems in e-commerce. In *Proceedings of the First ACM Conference on Electronic Commerce (EC-99), Denver, CO, USA, November 3-5, 1999*. ACM, 1999.

[2] Alejandro Valencia-Arias, Hernán Uribe-Bedoya, Juan David Gonzalez-Ruiz, Gustavo Sánchez Santos, Edgard Chapoñan Ramírez, and Ezequiel Martínez Rojas. Artificial intelligence and recommender systems in e-commerce. trends and research agenda. *Intell. Syst. Appl.*, 2024.

[3] Sebastian Lubos, Alexander Felfernig, and Markus Tautschnig. An overview of video recommender systems: state-of-the-art and research issues. *Frontiers Big Data*, 2024.

[4] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*. ACM, 2016.

[5] Shaina Raza and Chen Ding. News recommender system: a review of recent progress, challenges, and opportunities. *Artif. Intell. Rev.*, 2022.

[6] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. Personalized news recommendation: Methods and challenges. *ACM Trans. Inf. Syst.*, 2023.

[7] Mattia Giovanni Campana and Franca Delmastro. Recommender systems for online and mobile social networks: A survey. *Online Soc. Networks Media*, 2017.

[8] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[9] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, 2024.

[10] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. Deepseek-v3 technical report. *CoRR*, 2024.

[11] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily

Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, 2024.

[12] Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024: Student Research Workshop, St. Julian's, Malta, March 21-22, 2024*. Association for Computational Linguistics, 2024.

[13] Chengpeng Li, Guanting Dong, Mingfeng Xue, Ru Peng, Xiang Wang, and Dayiheng Liu. Dotamath: Decomposition of thought with code assistance and self-correction for mathematical reasoning. *CoRR*, 2024.

[14] Shima Imani, Liang Du, and Harsh Shrivastava. Mathprompter: Mathematical reasoning using large language models. In *Proceedings of the The 61st Annual Meeting of the Association for Computational Linguistics: Industry Track, ACL 2023, Toronto, Canada, July 9-14, 2023*. Association for Computational Linguistics, 2023.

[15] Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. Faithful logical reasoning via symbolic chain-of-thought. Association for Computational Linguistics, 2024.

[16] Xiaoxue Cheng, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. Think more, hallucinate less: Mitigating hallucinations via dual process of fast and slow thinking. *CoRR*, 2025.

[17] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*. ACM, 2018.

[18] Donghua Liu, Jing Li, Bo Du, Jun Chang, and Rong Gao. DAML: dual attention mutual learning between ratings and reviews for item recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. ACM, 2019.

[19] Jie Shuai, Kun Zhang, Le Wu, Peijie Sun, Richang Hong, Meng Wang, and Yong Li. A review-aware graph contrastive learning framework for recommendation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*. ACM, 2022.

[20] Alicia Tsai, Adam Kraft, Long Jin, Chenwei Cai, Anahita Hosseini, Taibai Xu, Zemin Zhang, Lichan Hong, Ed Huai-hsin Chi, and Xinyang Yi. Leveraging LLM reasoning enhances personalized recommender systems. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*. Association for Computational Linguistics, 2024.

[21] Yi Fang, Wenjie Wang, Yang Zhang, Fengbin Zhu, Qifan Wang, Fuli Feng, and Xiangnan He. Reason4rec: Large language models for recommendation with deliberative user preference alignment. *CoRR*, 2025.

[22] Jieyong Kim, Hyunseo Kim, Hyunjin Cho, SeongKu Kang, Buru Chang, Jinyoung Yeo, and Dongha Lee. Review-driven personalized preference reasoning with large language models for recommendation. *CoRR*, 2024.

[23] Zewen Long, Liang Wang, Shu Wu, Qiang Liu, and Liang Wang. Got4rec: Graph of thoughts for sequential recommendation. *CoRR*, 2024.

[24] Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Christopher Leung, Jiajie Tang, and Jiebo Luo. Llm-rec: Personalized recommendation via prompting large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*. Association for Computational Linguistics, 2024.

[25] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). In *RecSys*, 2022.

[26] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

[27] Wenjun Peng, Guiyang Li, Yue Jiang, Zilong Wang, Dan Ou, Xiaoyi Zeng, Derong Xu, Tong Xu, and Enhong Chen. Large language model based long-tail query rewriting in taobao search. In Tat-Seng Chua, Chong-Wah Ngo, Roy Ka-Wei Lee, Ravi Kumar, and Hady W. Lauw, editors, *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*. ACM, 2024.

[28] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian J. McAuley. Bridging language and items for retrieval and recommendation. *CoRR*, 2024.

[29] Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. Sft or rl? an early investigation into training r1-like reasoning large vision-language models, 2025.

[30] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. *CoRR*, 2025.

[31] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, 2024.

[32] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, 2025.

[33] OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea

Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

[34] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian J. McAuley. Bridging language and items for retrieval and recommendation. *CoRR*, 2024.

[35] Zhaopeng Qiu, Xian Wu, Jingyue Gao, and Wei Fan. U-BERT: pre-training user representations for improved recommendation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021.

[36] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*. The Association for Computer Linguistics, 2011.

[37] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning. In *NeurIPS*, 2022.

[38] Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. Quiet-star: Language models can teach themselves to think before speaking. *CoRR*, 2024.

[39] Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

[40] Lei Zheng, Vahid Noroozi, and Philip S. Yu. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*. ACM, 2017.

[41] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *RecSys*, 2023.

[42] Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. Collm: Integrating collaborative embeddings into large language models for recommendation. *CoRR*, abs/2310.19488, 2023.

[43] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, and Xiang Wang. Llara: Aligning large language models with sequential recommenders. *CoRR*, abs/2312.02445, 2023.

[44] Xiaoyu Kong, Jiancan Wu, An Zhang, Leheng Sheng, Hui Lin, Xiang Wang, and Xiangnan He. Customizing language models with instance-wise lora for sequential recommendation. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[45] Junguang Jiang, Yanwen Huang, Bin Liu, Xiaoyu Kong, Ziru Xu, Han Zhu, Jian Xu, and Bo Zheng. Large language models are universal recommendation learners. *CoRR*, abs/2502.03041, 2025.

[46] Yu Wang, Zhiwei Liu, Jianguo Zhang, Weiran Yao, Shelby Heinecke, and Philip S. Yu. DRDT: dynamic reflection with divergent thinking for llm-based sequential recommendation. *CoRR*, 2023.

[47] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *J. Artif. Intell. Res.*, 1996.

[48] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction, 2nd Edition*. MIT Press, 2018.

[49] Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *CoRR*, 2023.

[50] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, 2017.

[51] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 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*, 2023.

[52] Yuxin Chen, Junfei Tan, An Zhang, Zhengyi Yang, Leheng Sheng, Enzhi Zhang, Xiang Wang, and Tat-Seng Chua. On softmax direct preference optimization for recommendation. In *NeurIPS*, 2024.

[53] Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *CoRR*, 2025.

[54] Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, Hanghang Tong, and Heng Ji. Rm-r1: Reward modeling as reasoning, 2025.

[55] Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *CoRR*, 2025.

You are a helpful assistant. Your task is to analyze a user's purchase history, summarize their preferences, analyze a new target item, and then generate a recommendation rationale and predict a rating for that target item. Please follow these steps precisely:

1. ** User Interest Extraction :** For *each* item in the provided `user_purchase_history`, analyze the user's `rating` and `review`. Identify and state the points the user liked (`[like]`) and disliked (`[dislike]`). Then based on *all* the points summarized, consolidate the user's overall interest. Categorize them into positive (`[pos]`) and negative (`[neg]`) aspects. Adhere strictly to the following format:
   ``` <analyze user>…</analyze user> ```
2. ** Item Attributes Extraction :** Extract the features of the target item provided in `{target_item}`. Predict potential points the user might like (`[like]`) or dislike (`[dislike]`) about this specific item, considering general product attributes. Adhere strictly to the following format:
   ```< analyze item>… < /analyze item>```
3. **Match Analysis:** First, internally reason (`<match>`) about *why* the target item would (or would not) be a good recommendation for this user. This reasoning should explicitly connect the user's overall interest (from Step 1) with the target item's potential likes/dislikes (from Step 2). Provide detailed justifications. Adhere strictly to the following format:
   ``` <match>… </match> ```
4. ** Rating Prediction :** Then, provide the final predicted rating for the target item within the `<rate>` tags. The answer should *only* contain the predicted numerical rating (e.g., on a 1-5 scale). Adhere strictly to the following format:
   ``` <rate>… </rate>```

Your inputs will be `user_purchase_history` and ` target_item `. Ensure your output follows the specified formats and uses the `<analyze user>`, `<analyze item>`, `<match>` and `<rate>` tags correctly.

<user> user_purchase_history: {history}
target_item: {target_item }
</user>

Figure 5: System Prompt

A System Prompt

As illustrated in Fig 5, we guide the early outputs of the LLM through a structured process in the system prompt to achieve faster training convergence and superior model performance. Specifically, within the `<analyze user>` and `</analyze user>` tags, we directed the LLM to first list the features [like] and disliked features [dislike] of each product based on the user's historical interaction records, and then summarize the user's complete preferences using [pos] and [neg] tags. Subsequently, for the target item, we encourage the LLM to use [like] and [dislike] tags within the `<analyze item>` and `</analyze item>` tags to summarize the features that the user might like and dislike about the target item. Following this, the LLM engaged in a thoughtful analysis of the match between the user and the target item within the tags `<match></match>`, and finally provided the predicted user rating within the tags `<rate></rate>`.

B Limitation

While RecZero and RecOne exhibit promising results, this study is not without its limitations. First, due to computational constraints, we were unable to fully assess the potential performance gains that could be achieved by leveraging larger base models as the foundation for RL. This limitation may hinder a thorough evaluation of the models' true performance potential. Second, the study did not explore whether RecZero and RecOne could serve as viable replacements for existing Teacher models in generating cold-start data for multi-round self-iterative optimization. These limitations highlight the need for future work to investigate the scalability and efficacy of RecZero and RecOne in more resource-intensive and complex iterative settings, thereby providing a more comprehensive assessment of their practical applicability.

C Experimental Design and Evaluation

Datasets

To comprehensively validate the effectiveness of our proposed RecZero and RecOne, we conduct systematic experiments on four representative real-world recommendation datasets. Through comparative analysis with various baseline models, including traditional recommendation methods and state-of-the-art (SOTA) LLM-based approaches, we thoroughly demonstrate the superiority of our proposed methods. These datasets span across different domains and scenarios, specifically including: Book, Music and Yelp.

- **Book:**
  This dataset is derived from the "Book" subset of the widely used Amazon dataset for recommendation scenario evaluation. It records a large number of ratings, reviews, and rich book product metadata from users on the Amazon platform in the book scenario.
- **Music:**
  Similarly, this dataset is from the "Music" subset of the widely used Amazon dataset for recommendation scenario evaluation, containing user ratings and reviews in the music domain.
- **Yelp:**
  The Yelp Open dataset contains a large number of ratings and reviews from consumers for local restaurants and stores, and is widely used for performance evaluation in recommendation scenarios.

Baselines.

We compare RecZero and RecOne with a broad set of baselines that cover traditional collaborative filtering (CF) methods, review-based methods, and LLM-based approaches: MF, which is a classic collaborative filtering method; DeepCoNN, NARRE, and DAML, which rely on review text to learn richer user and item representations. DeepCoNN uses CNNs for joint modeling, NARRE applies attention to select informative reviews, and DAML models interactions between users and items. We also include Rec-SAVER, EXP3RT, and Reason4Rec, which are based on large language models. Rec-SAVER takes users' historical interactions and target item metadata as input to a teacher model to generate intermediate reasoning traces, which are distilled into a smaller student model to strengthen rating prediction. EXP3RT and Reason4Rec decompose a single step prediction into multiple reasoning steps, improving accuracy on reasoning enhanced rating prediction.

Training Protocol.

In our study, we employ the Qwen2.5-7B-Instruct-1M model as the starting point for RL due to its strong instruction-following capabilities and planning abilities acquired during pre-training. For traditional baseline experiments, we utilize a single H20 GPU, while the LLM-based baselines and our RecZero and RecOne frameworks are executed on an 8-card H20 GPU setup. All experiments are conducted using Python 3.9.

Evaluation Protocol.

For the rating prediction task, we employ Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as evaluation metrics. MAE measures the prediction accuracy of the model by calculating the average of the absolute errors between predicted values and true values, where a smaller value indicates better prediction performance. RMSE evaluates the model's predictive capability by computing the square root of the average of the squared errors between predicted values and true values. It is more sensitive to larger errors and provides a better reflection of the overall prediction stability of the model.

D Statistics

For both RecZero and RecOne, we utilize the Qwen2.5-7B-Instruct-1M model as the starting point for RL. During training, the batch size is set to 8, and the learning rate is 2e-6. Each data sample undergoes 8 rollouts during the training process. We set the sampling temperature to 1.0, the training epoch to 1, and the KL divergence to 0. Additionally, for the cold-start process of RecOne, we employ reasoning data provided by DeepSeek-R1 for cold-start initialization. We partition the dataset and select 1000 data samples that are not included in either the training or test sets for the cold-start experiments.