Preference Reasoning via Curriculum-based Reinforcement Learning for Review-based Recommendation

Anonymous ACL submission

Abstract

Recommendation has advanced by leveraging the reasoning capabilities of Large Language Models (LLMs). Existing approaches mainly rely on knowledge distillation, where a smaller student language model (LM) is trained using pseudo-labels generated by a teacher LLM. However, supervising an LM with a single reasoning text can limit its generalization ability, since a (user, item) interaction can be explained through various reasoning paths. To address this, we propose a **C**urriculum-driven **REA**soning op**T**imization f**O**r **R**ecommendation (**CREATOR**), a recommendation framework that optimizes the LM through reinforcement learning (RL) without fixed pseudo-labels. CREATOR employs a *self-paced reward curriculum (**SPRC**)* with two rewards: (R1) a rating-aligned reward for fine-grained preference modeling and (R2) a sentiment-aligned reward for early-stage stability. By adaptively adjusting the weights of these rewards based on the LM's learning progress, CREATOR enables the LM to transition from capturing general sentiment to reasoning about precise ratings. Experiments on three real-world datasets show that CREATOR consistently outperforms state-of-the-art competitors in rating prediction while providing more reliable and preference-aligned explanations. Our code and data are available at https://shorturl.at/8Dn7X.

1 Introduction

The goal of *recommender systems* is to determine a small set of items that users would prefer on large-scale e-commerce platforms. Early studies largely relied on users' past interaction histories to recommend items (Hwang et al., 2016; He et al., 2017; Chen et al., 2018; Wang et al., 2019; Lu et al., 2015; Zhang et al., 2019). To further improve recommendation accuracy, many studies have explored incorporating additional information, such as item multimodal features (Shen et al., 2013; Zhang et al., 2021; Kim et al., 2024), social networks (Konstas et al., 2009; Wu et al., 2019b; Geng et al., 2015), and review texts (Wu et al., 2019a; Shuai et al., 2022; Yang et al., 2023).

In this work, we focus on leveraging review texts. Since review texts describe user preferences and item characteristics in natural language, recent studies have increasingly adopted language models (LMs) due to their strong natural language understanding and *reasoning capabilities* (Geng et al., 2022; Ma et al., 2024; Tsai et al., 2024; Kim et al., 2025). Here, *reasoning* refers to the process of generating textual intermediate steps (called *reasoning text*) to solve a given problem; it enables the LM to achieve more-accurate solutions (Qiao et al., 2023; Shao et al., 2024; Guo et al., 2025). Recent studies leveraging LMs have explored methods aimed at reasoning about a user's preference for an item based on the reviews written by the user (*i.e.*, *user reviews*) and the reviews written for the item (*i.e.*, *item reviews*) (Tsai et al., 2024; Kim et al., 2025). Specifically, an LM generates a preference reasoning text that explains the user's preference for the item, and then the same LM or a separate recommendation model predicts a preference score (*e.g.*, a rating) by utilizing the generated reasoning text.

To fine-tune LMs to reliably perform preference reasoning for user–item pairs, (Tsai et al., 2024; Kim et al., 2025) adopt a *knowledge distillation* framework. They generate pseudo-label reasoning texts by using a large language model (LLM) as a teacher (*e.g.*, GPT (Achiam et al., 2023)), which produces high-quality outputs without fine-tuning but requires a high computational cost, based on user/item reviews and target ratings. Using these labels, they fine-tune an LM (*i.e.*, student) such as Llama (Touvron et al., 2023), which is typically smaller and more computationally-efficient than the teacher LLM. Since the teacher generates the reasoning texts (*i.e.*, pseudo labels) from user/item reviews while being conditioned to be

consistent with the target rating, the texts naturally include user/item information grounded in the reviews while aligning with the preference (*i.e.*, rating). Using these pseudo-labels, the student LM can be trained to generate preference-aligned reasoning texts grounded in user/item reviews.

However, for the same (user, item) pair, there can be many plausible reasoning texts; *i.e.*, there is *no clear criterion* for which one should be regarded as a valid pseudo-label. When we consider that the primary mission of recommendation systems is to understand user preferences, any reasoning text *consistent with the user–item preference* (*i.e.*, the rating) can be considered valid. Instead of relying on knowledge distillation with a *single* pseudo-label reasoning text, we fine-tune the LM via reinforcement learning (RL). Toward this end, we evaluate the quality of generated reasoning texts using the rating-prediction accuracy obtained with the texts and use it as a *reward* to optimize the LM, improving both rating prediction accuracy and the explanatory power with the generated reasoning texts. We design two rewards for RL, (R1) a rating-aligned reward and (R2) a sentiment-aligned reward, and propose a self-paced reward curriculum (SPRC) based on these two rewards.

- **(R1) Rating-aligned reward:** it evaluates the quality of generated reasoning texts based on the degree of rating prediction accuracy. Specifically, the rating-aligned reward assigns a higher value when the rating inferred from the reasoning text is closer to the ground-truth rating. This reinforces the LM to generate only the reasoning texts that support accurately predicting the user–item preference (*i.e.*, the rating), even though the same pair can be described by many different reasoning texts.

- **(R2) Sentiment-aligned reward:** however, in the *early stage* of RL, many generated reasoning texts receive similarly low rating-aligned rewards, making it difficult to distinguish their quality and thereby destabilizing training. To address this issue, we use a sentiment-aligned reward that assigns the maximum reward (1) when the sentiment of the inferred rating (*i.e.*, simply, positive or negative) matches the sentiment of the ground-truth rating, and the minimum reward (0) otherwise, providing a coarse, but clearer learning signal for the stability in early training.

- **Self-paced reward curriculum (SPRC):** while the sentiment-aligned reward contributes to training stability in the early stage, the rating-aligned reward does to fine-grained rating prediction. To combine the two rewards in a harmonious way, *SPRC* assigns a larger weight to the sentiment-aligned reward at the beginning and gradually increases the weight of the rating-aligned reward as the model performs better at maximizing the sentiment-aligned reward. This helps the LM to start the training stably in the early stage and generate reasoning texts that benefit more-accurate rating prediction in the later stage.

In this paper, we propose a **C**urriculum-driven **REA**soning op**T**imization f**O**r **R**ecommendation (**CREATOR**), a recommendation framework that learns to generate precise preference reasoning from review-based user and item profiles. CREATOR consists of three main stages: (S1) Profiling, (S2) Preference Reasoning, and (S3) Rating Prediction; note that our main contribution lies in (S2). In (S1), an **LLM** (*e.g.*, GPT) generates a textual profile that summarizes a user's (resp. an item's) *likes* and *dislikes* from the user's (resp. the item's) reviews. In (S2), an **LM**-based *preference reasoner* (*e.g.*, Llama) generates a preference reasoning text for a target (user, item) pair based on the user and item profiles generated in (S1). To this end, we fine-tune the preference reasoner using *SPRC* based on (R1) the *rating-aligned reward* and (R2) the *sentiment-aligned reward*, and apply group relative policy optimization (GRPO) (Shao et al., 2024) as the RL algorithm. In (S3), the reasoning generated by the preference reasoner in (S2) is leveraged by *a separate recommendation module* to predict a rating. To enable lightweight adaptation to the target domain, this stage uses a frozen pretrained LM encoder (*e.g.*, BGE) (Chen et al., 2024) with a fine-tunable multi-layer perceptron (MLP), taking only the generated reasoning text as input.

Our contributions are summarized as follows:

- **Motivation.** We posit that a user–item pair can be explained by different preference reasoning texts; we claim that using a single pseudo-label text to fine-tune an LM is inappropriate. Therefore, we suggest learning the LM *without pseudo-label texts* by evaluating whether the preference predicted from the generated reasoning text aligns with the true preference.

- **Novel Framework.** We propose CREATOR, a novel recommendation framework using RL for

accurate and explainable rating prediction.

  - To generate preference-aligned reasoning texts, CREATOR maximizes a rating-aligned reward and a sentiment-aligned reward that evaluate the rating prediction accuracy obtained from the reasoning text.
  - CREATOR adopts SPRC that emphasizes the sentiment-aligned reward to ensure stable RL in the early stage of training and gradually shifts to the rating-aligned reward to enable accurate rating prediction.

- **Extensive Experiments.** We conduct extensive experiments using three datasets and eight competitors. Our results validate that CREATOR consistently outperforms the LM-based competitors in *all cases* in accuracy, demonstrating the effectiveness of our reward-based optimization over the distillation-based methods.

2 Notations and Problem Definition

Notations. Let $\mathcal{U}$ and $\mathcal{V}$ denote the sets of users u_i and items v_j, respectively. For an interaction pair (u_i, v_j), x_{ij} and r_{ij} denote the review and rating given by user u_i to item v_j, where x_{ij} is a natural-language sequence of one or more sentences and $r_{ij} \in \{1, 2, 3, 4, 5\}$. We represent the set of observed interaction records as $\mathcal{I}$, denoting each record as $(u_i, v_j, x_{ij}, r_{ij})$. For a non-interaction pair (u_i, v_j), both x_{ij} and r_{ij} are unknown.

Problem Definition. Given observed interactions $\mathcal{I}$ and a target non-interaction pair (u_i, v_j), the goal of review-based recommendation is to predict the rating $\hat{r}_{ij}$ that user u_i would assign to item v_j.[1]

3 Proposed Method

In this section, we propose a novel LM-based recommendation framework, named CREATOR, that learns preference reasoning via RL. In Section 3.1, we present the overview of CREATOR. In Section 3.2, we describe the three main stages of CREATOR in detail. Finally, we describe the optimization procedures of CREATOR in Section 3.3.

3.1 Overview

Figure 1 illustrates the overall procedure of CREATOR. Given a target non-interaction pair (u_i, v_j), CREATOR predicts the rating $\hat{r}_{ij}$ through three stages: **(S1) Profiling, (S2) Preference Reasoning, and (S3) Rating Prediction.**

In (S1), we summarize the reviews written by the target user and those written for the target item to construct a user profile $\bar{u}_i$ and an item profile $\bar{v}_j$. In (S2), an LM-based preference reasoner generates a *preference reasoning text* $\bar{x}_{ij}$ from $\bar{u}_i$ and $\bar{v}_j$, and is fine-tuned with the SPRC based on (R1) a rating-aligned reward and (R2) a sentiment-aligned reward. In (S3), CREATOR encodes $\bar{x}_{ij}$ into $\bar{\mathbf{x}}_{ij}$ using a pretrained LM encoder (*e.g.*, BGE) and predicts $\hat{r}_{ij}$ with an MLP predictor.

3.2 Architecture

In this section, we describe the inference-time architecture of CREATOR.

3.2.1 Profiling (S1)

The goal of (S1) is to construct the user profile $\bar{u}_i$ and item profile $\bar{v}_j$ for a target pair (u_i, v_j). Since the same procedure is applied to users and items, we describe only the user side. To this end, we use an **LLM** (*e.g.*, GPT) to extract key preferences from each review x_{ik}, separating them into *likes* and *dislikes*, using the following prompt:

```
Your task is to analyze the following review and rating for {
      dataset} and extract the item preferences expressed by
      the user.

### Instructions:
1. Infer likes and dislikes from sentiment or tone of the
      review.
2. Include implied preferences even if they are not explicitly
       stated.
3. Include only item features or qualities (not user actions
      or decisions such as "would buy again").
4. If no preferences are mentioned, output an empty field
      after the colon.

### Input:
Review: {review}
Rating: {rating}

### Output (exactly two lines, and nothing else):
-Likes: <comma-separated list>
-Dislikes: <comma-separated list>
```

The summary s_{ik} is obtained by concatenating the extracted preferences with the title of item v_k.

Next, to construct the user profile $\bar{u}_i$, we concatenate the summaries s_{ik} of the reviews x_{ik} written to items v_k by user u_i. However, leveraging all summaries is computationally inefficient, so we construct $\bar{u}_i$ using up to the most recent H summaries. We get the item profile $\bar{v}_j$ in the same manner.

3.2.2 Preference Reasoning (S2)

The goal of (S2) is to generate the *preference reasoning text* $\bar{x}_{ij}$ given the user profile $\bar{u}_i$ and the item profile $\bar{v}_j$. To this end, we input $\bar{u}_i$ and $\bar{v}_j$ to

[1] Following prior review-based and LM-based recommendation studies (Liu et al., 2019; Kim et al., 2025; Fang et al., 2025), we focus on rating prediction in this work.

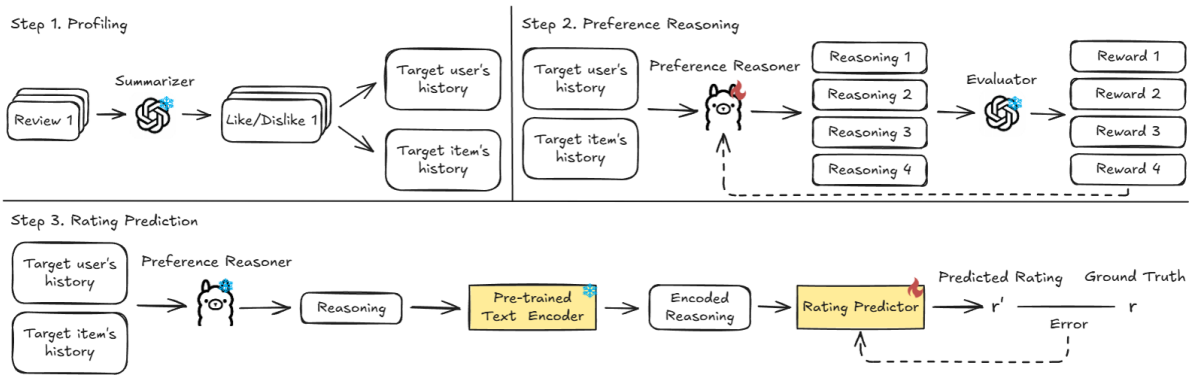


Figure 1: Overview of CREATOR composed of (S1) Profiling, (S2) Preference Reasoning, and (S3) Rating Prediction. In (S2), we fine-tune the preference reasoner with the SPRC based on (R1) the rating-aligned reward and (R2) the sentiment-aligned reward.

the preference reasoner with the following prompt:

```
You are given information about a user and a candidate {
      dataset} item, "{item_title}". The user's preferences
      are summarized by the user's review history, and the
      item's characteristics are summarized by reviews written
       by other users.

### User Review History on other items ###
{user_profile}

### Item Review History by other users ###
{item_profile}

Write a brief explanation (3-5 sentences) of whether the user
      would like the candidate item, based on the user's
      preferences and the item's features. DO NOT include a
      rating such as "the user would rate it X/5".
```

The preference reasoner is fine-tuned with RL using the SPRC to reliably generate $\bar{x}_{ij}$ aligned with user u_i's true preference, *i.e.*, the rating (see Section 3.3).

3.2.3 Rating prediction (S3)

The goal of (S3) is to predict $\hat{r}_{ij}$ from the generated reasoning text $\bar{x}_{ij}$ in (S2). We formulate this as a text regression task (Genkin et al., 2007; Wang et al., 2010, 2022) and encode $\bar{x}_{ij}$ into $\bar{\mathbf{x}}_{ij} \in \mathbb{R}^{d_1}$ using a pretrained sentence embedding model, BGE (Chen et al., 2024):

$$\bar{\mathbf{x}}_{ij} = \mathrm{BGE}\left(\bar{x}_{ij}\right), \tag{1}$$

where $\mathrm{BGE}(\cdot)$ denotes the sentence embedding produced by the BGE encoder, and d_1 represents the dimensionality of the embedding.

Next, we apply a multi-layer perceptron (MLP) to $\bar{\mathbf{x}}_{ij}$ to predict the rating $\hat{r}_{ij}$:

$$\hat{r}_{ij} = \sigma\left(\bar{\mathbf{x}}_{ij}\mathbf{w}_1\right)\mathbf{w}_2 + b_i + b_j, \tag{2}$$

where $\mathbf{w}_1 \in \mathbb{R}^{d_1 \times d_2}$ and $\mathbf{w}_2 \in \mathbb{R}^{d_2 \times 1}$ denote the learnable weights of the MLP predictor, and d_2 denotes its hidden dimensionality. $b_i \in \mathbb{R}$ and $b_j \in \mathbb{R}$ represent the learnable user bias and the item bias, respectively. $\sigma\left(\cdot\right)$ denotes an activation function, for which we employ SiLU (Elfwing et al., 2018). We keep the BGE frozen and fine-tune only the MLP, enabling fast training for rating prediction.

3.3 Optimization

In this section, we describe how we optimize CREATOR for preference reasoning and rating prediction. Due to the cost of RL fine-tuning, we optimize the preference reasoner on a sampled subset $\mathcal{I}_{\mathrm{RL}}$ of the training data, while the MLP predictor is trained on the full dataset.

3.3.1 Preference Reasoning (S2)

To improve the preference reasoner's capability, we apply GRPO (Shao et al., 2024), a widely accepted reinforcement learning (RL) method. GRPO samples multiple responses (*i.e.*, reasoning texts) from the policy (*i.e.*, the reasoner) for the same input and updates the policy based on the *relative reward differences* among these responses. This enables RL fine-tuning without requiring a separate critic model (Shao et al., 2024). Following GRPO, given a user–item pair (u_i, v_j), we sample N preference reasoning texts $\bar{x}_{ij}^{(n)}$ for $n \in \{1, \ldots, N\}$ and compute a reward for each $\bar{x}_{ij}^{(n)}$.

The goal of preference reasoning is to generate $\bar{x}_{ij}^{(n)}$ aligned with the ground-truth rating r_{ij}. To this end, we fine-tune the reasoner using two reward functions: (R1) a rating-aligned reward and (R2) a sentiment-aligned reward; (R1) is designed for fine-grained preference prediction, whereas (R2) provides a coarse but clear learning signal to stabilize training in the early stage. To compute the two rewards, we first use an LLM-based evaluator to infer a rating $\bar{r}_{ij}^{(n)}$ from the n-th generated reasoning text $\bar{x}_{ij}^{(n)}$ of (u_i, v_j). We use a frozen (*i.e.*, not fine-tuned) LLM-based evaluator because

it provides a fixed reward criterion and can flexibly interpret diverse reasoning texts without requiring task-specific evaluator training, thus reducing dependence on a particular reward-training dataset. The LLM evaluator is prompted as follows:

```
Read the following generated reasoning text carefully and
      infer the most likely rating the user would give
      (1.0~5.0, where 5.0 is the highest).

### Generated reasoning:
{reasoning}

Respond only with a single float between 1.0 and 5.0.
```

We empirically verify that the LLM can accurately evaluate the reasoning text (see Section 4).

We define a rating-aligned reward $w_r^{(i,j,n)}$ based on the squared error between the estimated rating $\bar{r}_{ij}^{(n)}$ and the ground-truth rating r_{ij} as follows:

$$w_r^{(i,j,n)} = \max\left(0, 1 - \frac{(\bar{r}_{ij} - r_{ij})^2}{\tau^2}\right), \tag{3}$$

where τ is a tolerance threshold for a rating prediction error; the reward is clipped to 0 when $|\bar{r}_{ij}^{(n)} - r_{ij}| \geq \tau$. $w_r^{(i,j,n)}$ takes a continuous value in $[0, 1]$ yielding 1 when $\bar{r}_{ij}^{(n)} = r_{ij}$ and 0 when the error exceeds the tolerance.

Next, a sentiment-aligned reward $w_s^{(i,j,n)}$ takes a discrete value in $\{0, 1\}$ and is defined as follows:

$$\begin{aligned} w_s^{(i,j,n)} &= \begin{cases} 1, & \text{if } f(r_{ij}) = f(\bar{r}_{ij}^{(n)}), \\ 0, & \text{otherwise}, \end{cases} \\ f(r_{ij}) &= \begin{cases} 1, & \text{if } r_{ij} > 3, \\ -1, & \text{otherwise}, \end{cases} \end{aligned} \tag{4}$$

where $f(\cdot)$ maps a rating to its sentiment by assigning 1 to positive ratings ($r_{ij} > 3$) and -1 otherwise (i.e., neutral/negative ratings $r_{ij} \leq 3$), following (Liu et al., 2022a,b; Kang et al., 2023; Xi et al., 2024).

After computing the rating-aligned reward $w_r^{(i,j,n)}$ and the sentiment-aligned reward $w_s^{(i,j,n)}$, we combine them to fine-tune the reasoner for rating-aligned preference reasoning *stably and precisely*. To this end, we adopt a *SPRC* that dynamically adjusts the weights of the two rewards according to the model's competence during training. Specifically, the SPRC combines $w_r^{(i,j,n)}$ and $w_s^{(i,j,n)}$ as follows:

$$\begin{aligned} w_c^{(i,j,n)} &= \lambda^{(\mathcal{B})} w_r^{(i,j,n)} + \left(1 - \lambda^{(\mathcal{B})}\right) w_s^{(i,j,n)}, \\ \lambda^{(\mathcal{B})} &= \frac{1}{N \times |\mathcal{B}|} \sum_{(u_i, v_j) \in \mathcal{B}} \sum_{n=1}^{N} w_s^{(i,j,n)}, \end{aligned} \tag{5}$$

where $\mathcal{B}$ represents a mini-batch of interaction pairs (u_i, v_j) and $w_c^{(i,j,n)}$ denotes the combined reward. $\lambda^{(\mathcal{B})}$ balances $w_r^{(i,j,n)}$ and $w_s^{(i,j,n)}$ by assigning a larger weight to $w_s^{(i,j,n)}$ when sentiment alignment is poor, thereby providing a clear learning signal for *stable training*. As sentiment alignment improves, $\lambda^{(\mathcal{B})}$ increases, gradually shifting the training signal toward $w_r^{(i,j,n)}$ for fine-grained rating alignment. $\lambda^{(\mathcal{B})}$ is recomputed for each mini-batch during the training, enabling the curriculum to adapt automatically as the training progresses.

Based on the combined reward $w_c^{(i,j,n)}$, CREATOR optimizes the following GRPO-based objective $\mathcal{L}_{\text{RL}}$:

$$\begin{aligned} \mathcal{L}_{\text{RL}} = -\mathbb{E}_{\substack{(u_i, v_j, x_{ij}, r_{ij}) \sim \mathcal{I}_{\text{RL}}, \\ \{\bar{x}_{ij}^{(n)}\}_{n=1}^{N} \sim \pi_\theta(\cdot | \bar{u}_i, \bar{v}_j)}} \frac{1}{N} \sum_{n=1}^{N} \Big[\min\Big(\rho_{ij}^{(n)} \hat{A}^{(i,j,n)}, \\ \text{clip}\left(\rho_{ij}^{(n)}, 1-\epsilon, 1+\epsilon\right) \hat{A}^{(i,j,n)} \Big) - \beta \mathbb{D}_{KL}\left(\pi_\theta || \pi_{\theta_{\text{old}}}\right) \Big], \end{aligned}$$

$$\rho_{ij}^{(n)} = \pi_\theta(\bar{x}_{ij}^{(n)} \mid \bar{u}_i, \bar{v}_j) / \pi_{\theta_{\text{old}}}(\bar{x}_{ij}^{(n)} \mid \bar{u}_i, \bar{v}_j),$$

$$\hat{A}^{(i,j,n)} = \frac{w_c^{(i,j,n)} - \text{mean}\left(\{w_c^{(i,j,m)}\}_{m=1}^{N}\right)}{\text{std}\left(\{w_c^{(i,j,m)}\}_{m=1}^{N}\right)}, \tag{6}$$

where π_θ represents the policy (*i.e.*, preference reasoner) and $\pi_{\theta_{\text{old}}}$ denotes the frozen reference policy from the previous update. $\rho_{ij}^{(n)}$ denotes the probability ratio of the sampled reasoning text $\bar{x}_{ij}^{(n)}$ under the current policy π_θ relative to the reference policy $\pi_{\theta_{\text{old}}}$. The clip$(\cdot)$ operator follows the standard PPO-style clipping (Schulman et al., 2017), truncating $\rho_{ij}^{(n)}$ into $[1-\epsilon,\ 1+\epsilon]$ with threshold ϵ. $\hat{A}^{(i,j,n)}$ is a group-normalized advantage computed from the combined rewards $\{w_c^{(i,j,m)}\}_{m=1}^{N}$ sampled for the same training pair (u_i, v_j), serving as a baseline without learning a value function. $\mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\theta_{\text{old}}})$ denotes the KL divergence (Hershey and Olsen, 2007; Shao et al., 2024) between the current policy and the reference policy, used as a regularizer to prevent overly large policy updates, and β controls its strength. $\mathcal{I}_{\text{RL}}$ denotes a subset of the full training set $\mathcal{I}$, obtained by sampling to reduce the computational cost of GRPO. To ensure that the preference reasoner can generate reasoning texts for all rating levels, we randomly select an equal number of samples for each rating (1–5), *i.e.*, $|\mathcal{I}_{\text{RL}}|/5$ samples per rating. We apply QLoRA (Dettmers et al., 2023) to efficiently optimize the preference reasoner.

3.3.2 Rating Prediction (S3)

Finally, we train the MLP predictor by minimizing the mean squared error (MSE) loss (Wang and

Table 1: Comparison results of CREATOR and eight competitors on three datasets. The symbol * denotes that p-values are below 0.05, indicating the differences are statistically significant.

Dataset	Metric	MF	DAML	RGCL	GPT-4.1	GPT-5	Rec-SAVER	EXP3RT	Reason4Rec	CREATOR
Music	RMSE	0.7929	0.7848	0.7665	1.0371	0.9982	0.9262	0.8691	0.7722	**0.7497***
	MAE	0.5903	0.5703	0.5357	0.7222	0.6900	0.6463	0.6144	0.5442	**0.5244***
Book	RMSE	0.8462	0.8481	0.8431	1.4010	1.3349	0.9451	0.9417	0.8376	**0.8172***
	MAE	0.6218	0.6211	**0.5887**	1.1565	0.9511	0.6676	0.6147	0.6110	0.5988
Yelp	RMSE	1.0363	1.0405	1.0375	1.3977	1.3200	1.1282	1.2311	1.0418	**0.9939***
	MAE	0.7964	0.7964	0.7698	1.1342	0.9810	0.8295	0.8306	0.7586	**0.7364***

Bovik, 2009) between the predicted rating $\hat{r}_{ij}$ and the ground-truth rating r_{ij} as follows:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{|\mathcal{I}|} \sum_{(u_i, v_j, x_{ij}, r_{ij}) \in \mathcal{I}} (r_{ij} - \hat{r}_{ij})^2 . \quad (7)$$

We optimize the predictor using the full training set $\mathcal{I}$. This encoder-based design avoids the high inference cost of LLM-based predictors while providing stable numerical regression, which is empirically analyzed in Appendix B.

4 Evaluation

In this section, we evaluate CREATOR by addressing the following evaluation questions (EQs):

(EQ1): Does CREATOR outperform state-of-the-art methods in recommendation accuracy and preference reasoning quality?

(EQ2): Is the SPRC based on (R1) and (R2) effective for improving recommendation accuracy?

(EQ3): Is reinforcement learning more effective than supervised fine-tuning for optimizing the LM's preference reasoning capability?

(EQ4): Is CREATOR robust to different LLMs used as the evaluator for reward calculation?

We also conduct additional experiments on the encoder-based rating predictor, computational cost comparison, parameter sensitivity (history length H for profiling and RL training sample size $|\mathcal{I}_{\text{RL}}|$), and case studies. Due to space limitations, the detailed results are reported in Appendix B.

4.1 Evaluation Setup

Datasets. Following (Fang et al., 2025), we conduct our experiments on three real-world datasets covering different domains: two Amazon 5-core datasets,[2] *i.e.*, Book and Digital Music (Music), and the Yelp dataset.[3] Due to the page limitation, we provide their details in Appendix A.

Competitors. To verify the effectiveness of CREATOR, we consider 1 representative conventional CF-based recommender (**MF** (Koren et al., 2009)), 2 review-based recommenders (**DAML** (Liu et al., 2019), **RGCL** (Shuai et al., 2022)), 2 few-shot LLMs (GPT-4.1, GPT-5), and 3 LM-based recommenders for preference reasoning (**Rec-SAVER** (Tsai et al., 2024), **EXP3RT** (Kim et al., 2025), **Reason4Rec** (Fang et al., 2025)) as our competitors.[4] Following (Shuai et al., 2022; Kim et al., 2025; Fang et al., 2025), we focus on rating prediction and evaluate the performance of CREATOR and competitors using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as accuracy metrics. All reported results are the averages of five independent runs conducted with different random seeds.

Implementation Details. We utilize GPT-5 mini[5] as the review summarizer and evaluator LLMs for CREATOR, while employing LLaMA-3-8B-Instruct[6] as the preference reasoner. For reasoning-based competitors fine-tuned via knowledge distillation (Rec-SAVER (Tsai et al., 2024), EXP3RT (Kim et al., 2025), Reason4Rec (Fang et al., 2025)), we consistently use GPT-5 mini as the teacher LLM and LLaMA-3-8B-Instruct as the student LM for a fair comparison. We present more details of the implementations in Appendix A.

4.2 Results

EQ1: Comparison with Eight Competitors. We compare the rating prediction accuracy of CREATOR with that of eight competitors in Table 1, where the best and second-best results are highlighted in **boldface** and underlines, respectively. CREATOR consistently outperforms all LM-based competitors across all datasets and metrics. While few-shot LLMs (GPT-4.1 and GPT-5) perform poorly without domain adaptation, SFT-based

[2] https://cseweb.ucsd.edu/jmcauley/datasets/amazon/links.html
[3] https://www.yelp.com/dataset
[4] We cover the CF-based and review-based recommenders that were compared in the papers of the LM-based recommenders (Kim et al., 2025; Fang et al., 2025).
[5] https://platform.openai.com/docs/models/gpt-5-mini
[6] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

Table 2: Rating prediction errors of an LLM based on the generated reasoning texts and the target review.

Dataset	Music		Book		Yelp	
Metric	RMSE	MAE	RMSE	MAE	RMSE	MAE
Rec-SAVER	0.9987	0.7550	0.9946	0.7865	1.2085	0.9571
EXP3RT	0.9781	0.7037	0.9749	0.7689	1.1668	0.8662
Reason4Rec	0.9601	0.7352	0.9558	0.7521	1.1702	0.8843
CREATOR	0.9247	0.6959	0.9440	0.7417	1.1557	0.8563
Target Review	**0.4029**	**0.2551**	**0.3658**	**0.2688**	**0.3847**	**0.2240**

methods (Rec-SAVER, EXP3RT, and Reason4Rec) improve the results but still remain inferior to CREATOR. These results suggest that our RL-based optimization helps CREATOR generate more preference-aligned reasoning than distillation-based approaches. CREATOR also outperforms all non-LM baselines, except for Book MAE where RGCL performs slightly better. Although RGCL is a strong baseline, it cannot explain its predictions, whereas CREATOR achieves strong accuracy while generating human-readable reasoning.

Next, we evaluate how well the generated reasoning texts align with the ground-truth preference (*i.e.*, rating). To isolate reasoning quality from the effect of different downstream predictors, we feed the reasoning texts generated by LM-based methods into an LLM evaluator (GPT-5 mini), which predicts a rating from each text. As a reference, we also report the results obtained by using the target review of the corresponding (user, item) pair as input. As shown in Table 2, using the target review yields the highest accuracy, indicating that the LLM evaluator can *reliably infer user preferences from review text.* Among the competitors not using the target review, CREATOR consistently performs best, showing that it produces reasoning texts more closely aligned with user preferences.

We further evaluate the diversity of the generated reasoning texts. For each LM-based method, we sample five reasoning texts for the same (user, item) pair with the sampling temperature fixed to 0.7, and compute the average pairwise cosine similarity (Reimers, 2019). Lower cosine similarity indicates greater diversity. Table 3 shows that CREATOR produces substantially more-diverse reasoning texts than other LM-based methods. Combined with Table 2, this indicates that CREATOR learns diverse reasonings while maintaining strong alignment with user preferences.

EQ2: Effectiveness of the SPRC. We use two rewards for reasoning optimization: (R1) a rating-aligned reward and (R2) a sentiment-aligned reward. CREATOR combines them with the SPRC

Table 3: Average pairwise cosine similarity of generated reasoning texts.

Dataset	Music	Book	Yelp
Rec-SAVER	0.9598	0.9449	0.9801
EXP3RT	0.9611	0.9507	0.9799
Reason4Rec	0.9584	0.9490	0.9767
CREATOR	**0.8867**	**0.8665**	**0.8918**

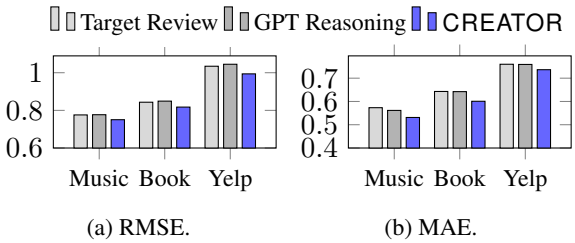
Table 4: Effectiveness of the proposed SPRC.

Dataset	Music		Book		Yelp	
Metric	RMSE	MAE	RMSE	MAE	RMSE	MAE
w/o R1	0.7674	0.5524	0.8410	0.6177	1.0088	0.7494
w/o R2	0.7619	0.5503	0.8401	0.6135	1.0088	0.7501
Two-stage	0.7613	0.5502	0.8342	0.6218	1.0067	0.7565
Fixed λ	0.7659	0.5398	0.8324	0.6360	1.0142	0.7576
Linear-decay	0.7622	0.5532	0.8328	0.6327	1.0157	0.7471
CREATOR	**0.7497**	**0.5244**	**0.8172**	**0.5988**	**0.9939**	**0.7364**

that adaptively adjusts their weights during training. To evaluate its effectiveness, we compare CREATOR with five variants: **w/o R1**, using only the sentiment-aligned reward (R2); **w/o R2**, using only the rating-aligned reward (R1); **Two-stage**, using the sentiment-aligned reward in the first half of training and the rating-aligned reward in the second half; **Fixed** λ, fixing λ at 0.5 (*i.e.*, $w_c = 0.5w_r + 0.5w_s$); and **Linear-decay**, linearly decreasing λ from 1 to 0. Table 4 presents the results. CREATOR achieves the best performance, followed by the variants that incorporate both rewards (Two-stage, Fixed λ, and Linear-decay). In contrast, the variants relying on only a single reward (w/o R1 and w/o R2) generally perform the worst. These findings suggest that both sentiment-aligned reward and the rating-aligned reward are indispensable for effective LM optimization. Moreover, adaptively combining the two rewards according to the model's pace yields better optimization than using fixed schedules.

EQ3: Comparison between RL and SFT. Instead of supervised fine-tuning (SFT) on fixed text labels, we optimize the preference reasoner via reinforcement learning (RL), allowing it to explore diverse reasoning texts. To verify the effectiveness of RL without fixed labels, we compare CREATOR with two SFT-based variants: **Target Review**, which is trained on the review of the target (user, item) pair as the ground-truth label, and **GPT Reasoning**, which is trained on a reasoning text generated by GPT-5 mini from the target (user, item) profile and the ground-truth rating.

As shown in Figure 2, CREATOR achieves the best performance, while Target Review and GPT Reasoning perform similarly but remain inferior to



(a) RMSE. (b) MAE.

Figure 2: Performance comparison of CREATOR with RL and its variant fine-tuned with SFT.

CREATOR. This suggests that the target review and GPT-generated reasoning provide similar supervision for the same $(user, item)$ pair. However, both SFT variants train the LM to reproduce a single text, which can limit coverage of valid reasoning expressions. In contrast, RL (*i.e.*, CREATOR) optimizes the LM with reward signals, allowing it to explore multiple preference-aligned reasoning texts and thereby improve generalization.

EQ4: Robustness to Evaluator Variations. Since CREATOR relies on an LLM-based evaluator to compute rewards, it is important to verify whether its effectiveness depends heavily on a specific evaluator. To this end, we compare CREATOR trained with different evaluator models. Specifically, we use GPT-5 mini as the default evaluator and further consider GPT-4.1 mini and GPT-5 as alternatives. Table 5 presents the results. Across all datasets and metrics, CREATOR shows highly similar performance regardless of which evaluator model is used. These results indicate that CREATOR is robust to variations in the evaluator model, and that its effectiveness does not depend on a particular evaluator choice.

Table 5: Rating prediction errors of CREATOR with different LLM evaluators.

Dataset	Music		Book		Yelp	
Metric	RMSE	MAE	RMSE	MAE	RMSE	MAE
GPT-4.1 mini	0.7514	0.5315	0.8146	**0.5972**	1.0021	0.7407
GPT-5 mini	**0.7497**	**0.5244**	0.8172	0.5988	**0.9939**	**0.7364**
GPT-5	0.7521	0.5264	**0.8134**	0.6055	0.9983	0.7379

5 Related Works

LM-based recommenders. Early studies used language models (LMs) to directly predict ratings or binary recommendations (Geng et al., 2022; Bao et al., 2023; Liu et al., 2023; Zhang et al., 2024). They explored prompt-based recommendation, efficient tuning, zero/few-shot recommendation, and the injection of collaborative filtering signals into LM inputs. More-recent studies leverage textual reviews to construct user and item profiles, generate preference reasoning for a target user–item pair, and use the generated reasoning for rating prediction (Tsai et al., 2024; Kim et al., 2025; Fang et al., 2025). To train LMs for this task, prior work mainly adopts knowledge distillation. Rec-SAVER (Tsai et al., 2024) and Reason4Rec (Fang et al., 2025) select teacher-generated reasoning candidates as pseudo-labels, while EXP3RT (Kim et al., 2025) directly generates rating-conditioned pseudo-label reasoning. However, a given (user, item) pair may admit multiple valid reasoning texts. Thus, imitating only a limited set of teacher outputs can restrict reasoning-space coverage and hurt generalization. To address this, we propose an RL-based framework that learns preference reasoning from reward signals instead of pseudo-labels.

Review-based recommenders. Review-based recommendation methods leverage textual reviews to better capture user preferences and item characteristics. Early work used neural architectures such as CNNs to encode reviews into latent user and item representations (Zheng et al., 2017), while later studies adopted attention mechanisms to highlight informative reviews (Chen et al., 2018; Liu et al., 2019). More-recent approaches further combine review signals with interaction graphs (Wu et al., 2019a; Gao et al., 2020; Shuai et al., 2022), with RGCL (Shuai et al., 2022) being a representative example. Despite their effectiveness, these methods do not explicitly model the reasoning process behind user preferences. By compressing reviews into latent embeddings, they offer limited interpretability, thus unable to provide human-readable explanations for recommendations.

6 Conclusion

In this work, we identify a limitation of existing LM-based approaches: relying on a single pseudo-label reasoning text for each (user, item) pair can hinder generalization. To address this, we propose CREATOR, an LM-based recommendation framework that fine-tunes a preference reasoner via reinforcement learning (RL) with reward signals instead of fixed pseudo-labels. CREATOR employs the self-paced reward curriculum (SPRC) with rating- and sentiment-aligned rewards, enabling stable training and accurate rating prediction. Extensive experiments show that CREATOR consistently outperforms eight competitors, and that both RL and the SPRC are effective for learning preference-aligned reasoning.

Limitations

Our current study leaves several directions for future work. First, CREATOR uses an external LLM evaluator to provide reward signals for preference reasoning. While the framework shows stable performance across multiple evaluator LLMs, relying on API-based evaluators incurs substantial financial overhead due to the repetition of paid API calls during training. Although encoder-based predictors are effective for the final rating prediction stage, they are difficult to apply directly as reward evaluators. Reward evaluation requires estimating the preference implied by a generated reasoning text, which we use as an indicator of preference-aligned reasoning quality. If trained only on review–rating pairs, an encoder-based evaluator may capture dataset-level rating patterns or user-specific rating tendencies, rather than reliably evaluating the preference signal expressed in generated reasoning texts. Developing such non-API evaluators would improve the practicality of our framework.

Second, we evaluate reasoning quality mainly through rating alignment and diversity. While these metrics are useful for measuring whether the generated reasoning is preference-relevant and meaningful, they do not fully capture other desirable properties such as usefulness, faithfulness, or readability from a human perspective. A broader evaluation with human judgments would provide a complete analysis of generated reasoning texts.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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 ACM International Conference on Recommender Systems (ACM Recsys)*, pages 1007–1014.

Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the ACM International Conference on World Wide Web (ACM WWW)*, pages 1583–1592.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, pages 10088–10115.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, pages 3–11.

Yi Fang, Wenjie Wang, Yang Zhang, Fengbin Zhu, Qifan Wang, Fuli Feng, and Xiangnan He. 2025. Reason4rec: Large language models for recommendation with deliberative user preference alignment. *arXiv preprint arXiv:2502.02061*.

Jingyue Gao, Yang Lin, Yasha Wang, Xiting Wang, Zhao Yang, Yuanduo He, and Xu Chu. 2020. Set-sequence-graph: A multi-view approach towards exploiting reviews for recommendation. In *Proceedings of the ACM International Conference on Information and Knowledge Management (ACM CIKM)*, pages 395–404.

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

Xue Geng, Hanwang Zhang, Jingwen Bian, and Tat-Seng Chua. 2015. Learning image and user features for recommendation in social networks. In *Proceedings of the IEEE International Conference on Computer Vision (IEEE ICCV)*, pages 4274–4282.

Alexander Genkin, David D Lewis, and David Madigan. 2007. Large-scale bayesian logistic regression for text categorization. *Technometrics*, pages 291–304.

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, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the ACM International Conference on World Wide Web (ACM WWW)*, pages 173–182.

John R Hershey and Peder A Olsen. 2007. Approximating the kullback leibler divergence between gaussian mixture models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE ICASSP)*, pages 317–320.

Won-Seok Hwang, Juan Parc, Sang-Wook Kim, Jongwuk Lee, and Dongwon Lee. 2016. "told you i didn't like it": Exploiting uninteresting items for effective collaborative filtering. In *Proceedings of the IEEE International Conference on Data Engineering (IEEE ICDE)*, pages 349–360.

Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed Chi, and Derek Zhiyuan Cheng. 2023. Do llms understand user preferences? evaluating llms on user rating prediction. *arXiv preprint arXiv:2305.06474*.

Jieyong Kim, Hyunseo Kim, Hyunjin Cho, SeongKu Kang, Buru Chang, Jinyoung Yeo, and Dongha Lee. 2025. Review-driven personalized preference reasoning with large language models for recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 1697–1706.

Yungi Kim, Taeri Kim, Won-Yong Shin, and Sang-Wook Kim. 2024. Monet: Modality-embracing graph convolutional network and target-aware attention for multimedia recommendation. In *Proceedings of the ACM International Conference on Web Search and Data Mining (ACM WSDM)*, pages 332–340.

Ioannis Konstas, Vassilios Stathopoulos, and Joemon M Jose. 2009. On social networks and collaborative recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 195–202.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, pages 30–37.

Donghua Liu, Jing Li, Bo Du, Jun Chang, and Rong Gao. 2019. Daml: Dual attention mutual learning between ratings and reviews for item recommendation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (ACM SIGKDD)*, pages 344–352.

Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149*.

Weiming Liu, Xiaolin Zheng, Mengling Hu, and Chaochao Chen. 2022a. Collaborative filtering with attribution alignment for review-based non-overlapped cross domain recommendation. In *Proceedings of the ACM International Conference on World Wide Web (ACM WWW)*, pages 1181–1190.

Weiming Liu, Xiaolin Zheng, Jiajie Su, Mengling Hu, Yanchao Tan, and Chaochao Chen. 2022b. Exploiting variational domain-invariant user embedding for partially overlapped cross domain recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 312–321.

Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. 2015. Recommender system application developments: a survey. *Decision support systems*, pages 12–32.

Qiyao Ma, Xubin Ren, and Chao Huang. 2024. Xrec: Large language models for explainable recommendation. *arXiv preprint arXiv:2406.02377*.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with language model prompting: A survey. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5368–5393.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

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*.

Jialie Shen, Meng Wang, Shuicheng Yan, and Peng Cui. 2013. Multimedia recommendation: technology and techniques. In *Proceedings of the International ACM SIGIR conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 1131–1131.

Jie Shuai, Kun Zhang, Le Wu, Peijie Sun, Richang Hong, Meng Wang, and Yong Li. 2022. A review-aware graph contrastive learning framework for recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 1283–1293.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Alicia 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. In *Findings of the Association for Computational Linguistics (ACL findings)*, pages 13176–13188.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the*

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD), pages 783–792.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 165–174.

Yuxia Wang, Daniel Beck, Timothy Baldwin, and Karin Verspoor. 2022. Uncertainty estimation and reduction of pre-trained models for text regression. *Transactions of the Association for Computational Linguistics*, pages 680–696.

Zhou Wang and Alan C Bovik. 2009. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*.

Chuhan Wu, Fangzhao Wu, Tao Qi, Suyu Ge, Yongfeng Huang, and Xing Xie. 2019a. Reviews meet graphs: Enhancing user and item representations for recommendation with hierarchical attentive graph neural network. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4884–4893.

Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019b. A neural influence diffusion model for social recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 235–244.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards open-world recommendation with knowledge augmentation from large language models. In *Proceedings of the ACM Conference on Recommender Systems (ACM Recsys)*, pages 12–22.

Wei Yang, Tengfei Huo, Zhiqiang Liu, and Chi Lu. 2023. Review-based multi-intention contrastive learning for recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 2339–2343.

Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. 2021. Mining latent structures for multimedia recommendation. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, pages 3872–3880.

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys*, pages 1–38.

Yang Zhang, Keqin Bao, Ming Yan, Wenjie Wang, Fuli Feng, and Xiangnan He. 2024. Text-like encoding of collaborative information in large language models for recommendation. *arXiv preprint arXiv:2406.03210*.

Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the ACM International Conference on Web Search and Data Mining (ACM WSDM)*, pages 425–434.

A Evaluation Setup

Table 6: Statistics of the three datasets. **Avg.** $|x_{ij}|$ denotes the average review length (in characters)

Dataset	#Users	#Items	#Train	#Valid	#Test	Avg. $\lvert x_{ij}\rvert$
Music	4,183	2,660	43,071	3,271	1,296	1,140
Book	25,265	24,113	163,218	17,013	18,351	703
Yelp	8,453	13,426	51,497	4,757	4,328	605

Dataset. We conduct our experiments on three public real-world datasets covering different domains: two Amazon 5-core datasets,[7] *i.e.*, Book and Digital Music (Music), and the Yelp dataset.[8] Since the Book and Yelp datasets are excessively large, we use their subsets following (Fang et al., 2025), while using the entire dataset for Music. Specifically, we utilize the last 12 months of interactions for Books and the last six months for Yelp. The training, validation, and test sets are split chronologically to prevent information leakage; we ensure that the validation and test sets do not contain cold-start users, following (Kim et al., 2025; Fang et al., 2025). Table 6 summarizes the statistics of these datasets.

Competitors. To verify the effectiveness of CREATOR, we consider 1 representative conventional CF-based recommender, 2 review-based recommenders, 2 few-shot LLMs, and 3 LM-based recommenders for preference reasoning as our competitors.[9]

- **MF** (Koren et al., 2009): a classic CF-based recommender that predicts the ratings of non-interacted (user, item) pairs by utilizing only the rating matrix.
- **DAML** (Liu et al., 2019): a review-based recommender that enhances user and item features by modeling the correlation between

[7] https://cseweb.ucsd.edu/~jmcauley/datasets.html
[8] https://www.yelp.com/dataset
[9] We cover the CF-based and review-based recommenders that were compared in the papers of the LM-based recommenders (Kim et al., 2025; Fang et al., 2025).

user reviews and item reviews through local and mutual attention mechanisms.

- **RGCL** (Shuai et al., 2022): a review-based recommender that enriches user and item features by incorporating reviews and ratings into user-item interaction graph modeling. It further incorporates contrastive learning based on node and edge discrimination as an auxiliary objective to the recommendation task.

- **LLM w/o fine-tuning** (Liu et al., 2023): a prompt-based approach that predicts ratings by leveraging the zero/few-shot capabilities of LLMs. In our experiments, we utilize few-shot prompts and employ GPT-4.1[10] and GPT-5[11] as the backbone LLMs.

- **Rec-SAVER** (Tsai et al., 2024): an LM-based recommender that leverages knowledge distillation for preference reasoning. A teacher LLM generates multiple reasoning paths based on user history and item metadata, and the student LM distills the label reasoning selected through self-verification.

- **EXP3RT** (Kim et al., 2025): an LM-based recommender that leverages knowledge distillation for preference reasoning. It constructs user and item profiles using reviews, and a teacher LLM generates the pseudo-labels for the reasoning texts based on these profiles, which are then distilled into a student LM.

- **Reason4Rec** (Fang et al., 2025): an LM-based recommender that performs explicit reasoning before predicting ratings. It employs three collaborative experts to execute a step-wise reasoning process consisting of preference distillation, preference matching, and feedback prediction by utilizing user reviews.

Implementation Details. We utilize GPT-5 mini[12] as the review summarizer and evaluator LLMs for CREATOR, while LLaMA-3-8B-Instruct[13] is employed as the preference reasoner. Although GPT-5 exhibits superior performance in text summarization and classification, we select GPT-5 mini for CREATOR to balance performance and computational cost. For reasoning-based competitors that involve fine-tuning an LM using knowledge distillation (Rec-SAVER (Tsai et al., 2024), EXP3RT (Kim et al., 2025), Reason4Rec (Fang et al., 2025)), we consistently use GPT-5 mini as the teacher LLM and LLaMA-3-8B-Instruct as the student LM for a fair comparison.

We employ pretrained-BGE [14] as an encoder-based LM for rating prediction, with the hidden dimensions d_1 and d_2 of the MLP predictor set to 1024 and 64, respectively. Across all datasets, τ is set to 2, the history length H for users and items is set to 10, the GRPO training sample size $|\mathcal{I}_{\text{RL}}|$ is 2000, and the GRPO group size N is 8. We set the clipping threshold ϵ to 0.2 and KL coefficient β to 0.002. We utilize QLoRA for fine-tuning the reasoning LM, and all learnable parameters are optimized using the Adam optimizer. All experiments are conducted on an Ubuntu 22.04 server equipped with NVIDIA RTX A6000 GPUs (48GB VRAM) and 256GB RAM, implemented using PyTorch 2.8.0.

Table 7: Comparison of rating prediction error and inference time for CREATOR using different predictors

Dataset	Metric	LLM w/o FT	LM w/ FT	ENC+MLP
Music	RMSE	0.9537	0.8076	**0.7497**
	MAE	0.6875	0.5980	**0.5244**
	Time (sec)	**17**	285	180
Book	RMSE	0.9544	0.8508	**0.8172**
	MAE	0.7269	0.7056	**0.5988**
	Time (sec)	**267**	3,828	1,035
Yelp	RMSE	1.0955	1.0440	**0.9939**
	MAE	0.8766	0.9355	**0.7364**
	Time (sec)	**65**	994	210

B Additional Results

B.1 Encoder-based Rating Predictor

Rather than using generative LMs for rating prediction, we utilize a predictor composed of a frozen encoder-based LM and an MLP (ENC+MLP) to efficiently predict ratings based on generated reasoning texts. To evaluate the effectiveness of this architecture, we compare its performance against two baselines: (1) a non-fine-tuned LLM-based predictor (LLM w/o FT) and (2) a fine-tuned LM-based predictor (LM w/ FT). Specifically, we employ the GPT-5 mini for LLM w/o FT, while LLaMA3-8b is selected for LM w/ FT. Both variants are designed to predict ratings by taking the user and

[10] https://platform.openai.com/docs/models/gpt-4.1
[11] https://platform.openai.com/docs/models/gpt-5
[12] https://platform.openai.com/docs/models/gpt-5-mini
[13] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[14] https://huggingface.co/BAAI/bge-m3

Table 8: Comparison of fine-tuning time for CREATOR and LM-based competitors

Method	Metric	Music	Book	Yelp
Rec-SAVER	**Time (hour)**	30	56	24
	RMSE	0.9262	0.9451	1.1282
EXP3RT	**Time (hour)**	37	108	46
	RMSE	0.8691	0.9417	1.2311
Reason4Rec	**Time (hour)**	54	194	75
	RMSE	0.7722	0.8376	1.0418
CREATOR	**Time (hour)**	**25**	**20**	**17**
	RMSE	**0.7497**	**0.8172**	**0.9939**

item profiles and the reasoning text as input, following (Kim et al., 2025; Fang et al., 2025). LM w/ FT is fine-tuned to directly predict the ground-truth ratings. LLM w/o FT and LM w/ FT are evaluated with an inference batch size of 8. For ENC+MLP, the encoder is run with a batch size of 1, while the MLP is executed with a batch size of 1024.

Table 7 presents the results. We compare the rating prediction accuracy as well as the inference time of the variants for each test set. In terms of accuracy, ENC+MLP performs best, followed by LM w/ FT, while LLM w/o FT shows the lowest performance. For inference latency, LLM w/o FT is the fastest, followed by ENC+MLP, whereas LM w/ FT is the slowest. While LLM w/o FT is fast, it is not suitable as a predictor in practice due to its API cost and limited domain adaptation. Although LM w/ FT leverages domain adaptation and outperforms LLM w/o FT, it remains inferior to the encoder+MLP regressor. This suggests that generative LMs are less suited for precise regression and harder to optimize for numerical accuracy. LM w/ FT also incurs substantially higher latency. Overall, ENC+MLP achieves strong accuracy with reasonable latency, supporting the use of an encoder-based LM for efficient and effective rating prediction.

B.2 Computational Cost

We utilize GRPO as our reinforcement learning algorithm to train CREATOR. To fine-tune CREATOR in a reasonable time, we use only a subset $\mathcal{I}_{\text{RL}}$ of the entire training samples ($< 5\%$) for reinforcement learning. We measure the *fine-tuning time for reasoning* of CREATOR and those for other LM-based competitors (*i.e.*, Rec-SAVER, EXP3RT, and Reason4Rec) to validate the efficiency of CREATOR.

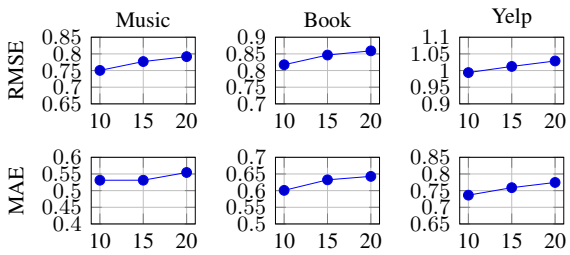


Figure 3: The effect of H on the accuracies of CREATOR.

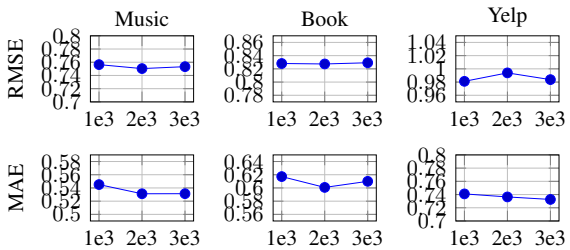


Figure 4: The effect of $|\mathcal{I}_{\text{RL}}|$ on the accuracies of CREATOR.

Table 8 reports the fine-tuning time (hours) and the rating prediction error (RMSE) of CREATOR and the competitors. By training the reasoner on a small RL subset, CREATOR reduces the reasoning fine-tuning time by approximately 36% compared to the fastest LM-based competitor. The reasoning fine-tuning time of CREATOR slightly increases from Yelp to Book to Music, consistent with their average review lengths (605, 703, and 1,140 characters, respectively). This is because longer reviews lead to longer user/item profiles, increasing the input length and thus the computational cost. Overall, these results suggest that CREATOR can learn effective preference reasoning with a small RL training subset, making it more practical for real-world applications.

B.3 Parameter Sensitivity

We utilize user's (resp. item's) H most recent reviews to construct the user profile (resp. item profile). We also sample a subset $\mathcal{I}_{\text{RL}}$ from the training set for RL. To examine the impact of H and the size of $\mathcal{I}_{\text{RL}}$ on CREATOR, we conduct a hyperparameter sensitivity analysis. Specifically, we compare the rating prediction performance by varying H within the range of $\{10, 15, 20\}$ and $|\mathcal{I}_{\text{RL}}|$ within the range of $\{1000, 2000, 3000\}$.

Figure 3 shows the effect of the history length H. For most datasets and metrics, performance remains similar or even slightly degrades when H is increased beyond 10. This suggests that the

Table 9: Reasoning texts for user *AVEOVIRYPMCSN* and book *Servants of the Storm*, along with the target review

Target Review: I **DNF** this one. It was just so boring! With a cover like this I expected **thriller and fast paced**, but what I got was **very wordy and incredibly slow**. I am not sure if it ever picked up and if it did I would love to know because I think I would have loved the story. I just couldn't sit there any longer reading **description after description**. Great idea, but the story needed to move way faster than it did.
CREATOR: Although the user appreciates psychological thrillers and layered character backstories, they may not connect with the southern fiction backdrop and the overly wholesome "boy next door" type featured in Servants of the Storm. In addition, the user **tends to lose interest in stories with sluggish momentum** or narrative gaps, which the book's underground demon storyline may introduce. As a result, the **mismatch in tone, pacing**, and character appeal suggests the user **would likely feel underwhelmed by this book**.
Reason4Rec: Based on the user's review history, they favor engaging, well-developed stories with strong character development, and thought-provoking themes. They appreciate atmospheric settings, complex plots, and stories that offer satisfying resolutions. The reviews for "Servants of the Storm" highlight its engaging supernatural elements, atmospheric setting, and strong protagonist, aligning with the user's preference for immersive worlds and character strength. However, some reviews mention **pacing issues** and a lack of character connection, which may pose a concern given the user's appreciation for clarity, character development, and emotional engagement. Overall, if the user values atmospheric, mature, and supernatural stories with strong characters, they are **likely to enjoy** "Servants of the Storm."

10 most recent reviews are sufficient to construct informative user and item profiles. In contrast, incorporating more reviews may make the context unnecessarily long, making it difficult to focus on the important information.

Figure 4 shows the effect of the RL subset size $|\mathcal{I}_{\text{RL}}|$. When $|\mathcal{I}_{\text{RL}}| = 1000$, the performance slightly degrades; however, the results for $|\mathcal{I}_{\text{RL}}| = 2000$ and 3000 are almost identical. This suggests that 2,000 samples are sufficient for GRPO training in our problem context, since sampling $N(= 8)$ reasoning texts per (user, item) pair enables the model to explore a diverse set of candidate reasonings and learn a reliable preference reasoning policy.

B.4 Case Study

We qualitatively compare the reasoning texts generated by LM-based methods: CREATOR and Reason4Rec, which achieves the highest rating prediction accuracy among the LM-based competitors. Table 9 presents the ground-truth review for the target user and item, along with the reasoning texts generated by the two methods. First, CREATOR accurately identifies why the user disliked the item and produces a final impression consistent with the target review (highlighted in green). In contrast, although Reason4Rec generates a lengthy reasoning text, only a small portion aligns with the user's preferences, and its final impression is incorrect (highlighted in red). Overall, CREATOR produces more-compact reasoning texts while being better aligned with the user's core preferences.

C Ethical Considerations

Potential Risks. CREATOR is intended to improve preference reasoning for recommendation, but several considerations remain important for real-world use. First, if the generated reasoning text is not fully faithful to the user's true preference, it may provide plausible but misleading explanations for recommendations and over-influence user decisions. Second, because the framework leverages user review histories and may rely on external LLM services for summarization or reward evaluation, practical deployment should consider privacy and data-governance issues when handling reviews written by users. Finally, because the framework learns from historical user reviews, it may reflect existing biases in the data, such as skewed item exposure or biased language in user-written reviews. These risks should be considered carefully in deployment-oriented settings, together with privacy safeguards and broader human-centered evaluation.

Artifacts and Terms of Use. We use publicly available datasets, pretrained models, and API services under their respective licenses or terms of use. All artifacts are used solely for research purposes in this work.

Dataset and Privacy Considerations. We use publicly released review datasets and do not perform additional data collection. We report aggregate dataset statistics and include only a limited case study for qualitative analysis.

D GenAI Usage Disclosure

The authors used ChatGPT to improve the academic tone of the manuscript and to check for typographical and grammatical errors. All scientific content, claims, and final writing were reviewed and finalized by the authors, who take full responsibility for the final manuscript.