From Thoughts to Tastes: Modeling Preference Hierarchies for Human-Aligned Sequential Recommendations

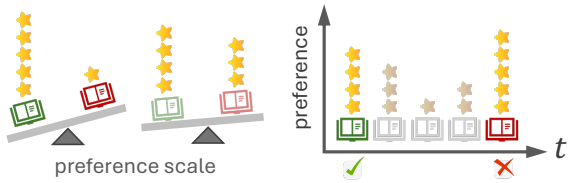
Qianlong Wen[1]* **and Zhongyu Ouyang**[2]* **and Chunhui Zhang**[2] **and Soroush Vosoughi**[2]† and **Yanfang Ye**[1]†
[1]University of Notre Dame [2]Dartmouth College

Abstract

Sequential recommendation systems aim to profile users via their interaction histories, akin to the human cognitive process to infer intent from observed behaviors. However, conventional large language model (LLM)-based recommender often fail to replicate the nuanced adaptability of human reasoning—specifically, the ability to contextualize preference hierarchies, personalized scales, and situational factors. Existing methods rely on reductive pairwise preference ranking, neglecting the multi-dimensional, dynamic nature of human decision-making and limiting alignment with cognitively grounded user profiling. In this paper, we propose RecPO, a preference optimization method that integrates contextual calibration into LLM-based recommenders to emulate human adaptive reasoning. Our framework employs adaptive reward margins that dynamically adjust preference signals by incorporating explicit user ratings and interaction latency, achieving granular alignment with individual decision patterns. Through extensive experiments, we underscore hierarchical preference modeling's role in bridging algorithmic recommendations and human cognitive strategies. RecPO not only surpasses state-of-the-art baselines by a significant margin but also uniquely balances temporal consistency (chronologically prioritizing preferred items) with avoidance of dispreferred items in future interactions. Code: https://anonymous.4open.science/r/RecPO-D3DB/

1 Introduction

In modern social media, recommender systems are ubiquitous in shaping user experiences by delivering personalized content across various online platforms, including e-commerce (Sarwar et al., 2000; Schafer et al., 1999), video streaming services (Davidson et al., 2010; Deldjoo et al., 2020), and social media networks (Ma et al., 2008; Jamali and Ester, 2010; Fan et al., 2019). These systems learn user preferences by capturing patterns in historical interactions between users and items, such as click-streams, merchandise purchase histories, or movie reviews. Sequential recommendation, as a specialized form of recommendation, predicts the next item a user is likely to interact with based on his/her historical behaviors. This form of recommendation plays a crucial role in real-world applications, such as movie recommendation on streaming platforms like Netflix, or curating the next song on Spotify based to listening history. Unlike static recommender systems, a sequential recommender is required to identify temporal dynamics and contextual nuances in user behavior sequences, where preferences are continuously evolving.



(i) Variance in Δpreference *(ii)* Time-discounted preference

Figure 1: Cognitive behaviors in decision-making: (i) Preference differences vary in constructed pairwise preference data; (ii) Preference is discounted by time.

Classical sequential recommender systems (Sun et al., 2019; Tang and Wang, 2018; Chang et al., 2021) leverage deep sequential architectures, such as recurrent neural networks (RNNs (Hidasi, 2016)) and Transformers (Kang and McAuley, 2018), to model user preferences from chronologically ordered interaction sequences. Information collected from platforms, including user/item IDs and contextual features (e.g., item descriptions, interaction timestamps), are encoded as low-dimensional vectors, which are then fed into these architectures for sequential preference pattern modeling and final prediction. These models are specifically tailored for sequential recommendation, thus highly effec-

*Equal contribution
†Corresponding author

tive with superior compactness and efficiency. On the other hand, recent advancements in foundation language models (LMs), especially large language models (LLMs), have stimulated growing interest in their application to various tasks, including sequential recommendation (Harte et al., 2023; Li et al., 2023; Yang et al., 2024; Bao et al., 2023; Zhang et al., 2023). With vast world knowledge and sophisticated reasoning capabilities, LLM-based recommenders hold substantial potentials for improving sequential recommendation with deep contextual understanding and complex user profiling.

User profiling plays a pivotal role in recommendation tasks by mirroring the advanced cognitive ability humans exhibit when adapting interpretations to individual perspectives during decision-making (Astington and Jenkins, 1995). However, current LLM-based recommender systems often struggle in achieving comparable cognitive flexibility. While existing preference modeling approaches, such as direct preference optimization (DPO) (Rafailov et al., 2024) and its derivatives (Chen et al., 2024; Meng et al., 2024; Amini et al., 2024), leverage pairwise preference data to rank preferred items over dis-preferred ones, this strategy oversimplifies the complexity of human preferences. Specifically, it neglects the hierarchical structure of user preferences, where the degree of preference (e.g., mild vs. strong) encodes critical user-specific signals, as illustrated in Figure 1(i). Moreover, previous efforts fail to account for time-discounted preferences. In sequential recommendation scenarios, for instance, suggesting a 4-star item in the near term holds greater practical value than proposing a 5-star item that may only become relevant in the distant future. This time-discounted aspect of human decision-making, depicted in Figure 1(ii), remains unaddressed in existing LLM-based recommender systems, limiting their alignment with real-world user behavior.

To bridge this gap: *First*, we conduct a proof-of-concept experiment to demonstrate the benefits of leveraging comprehensive user preference feedback in recommendation. *Then*, we propose RecPO, a framework that enhances LLM-based recommender systems by emulating the human cognitive process of contextual calibration through fine-grained preference feedback. Compared to previous methods, our framework is highlighted in the following aspects: *(i)* Instead of removing historical interactions with negative feedback from conditioned historical sequences, we retain interactions of all preference levels; *(ii)* Fine-grained explicit preference feedback or their proxy is provided; *(iii)* Time-discounted preference is considered when generating preference data; *(iv)* Preference alignment is tailored with an adaptive reward margin indicating gaps in preference levels. *Finally*, we conduct extensive experiments to show that our framework better aligns with human preference behaviors. It not only ensures that the most preferred item consistently ranks highest but also organizes items according to fine-grained preference levels. Moreover, it accurately relegates truly disliked items to lower ranks, capturing a nuanced understanding of user preferences and aversions.

2 Related Work

Sequential Recommendation Sequential recommendation models temporal user preferences in interaction sequences. Early methods adopt structures such as recurrent neural networks (GRU4Rec (Hidasi, 2016)) and self-attention mechanisms (SASRec (Kang and McAuley, 2018)). Recent advances integrate LLMs for their rich semantic understanding and contextual reasoning capabilities. These recommenders exploit textual information such as item descriptions as interaction histories for nuanced user profiling and interpretable predictions (Liao et al., 2024; Bao et al., 2023). Emerging approaches also utilize prompting (Geng et al., 2022) and multi-modal data (Yuan et al., 2023) for more comprehensive recommendation.

LLM Preference Alignment LLM alignment techniques ensure that models produce outputs aligned with human preferences and, have inspired significant advancements beyond general-purposes tasks such as recommendations. Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) and DPO (Rafailov et al., 2024) suggest fine-tuning based on human preference data. Building on DPO, methods like IPO (Azar et al., 2024), CPO (Xu et al., 2024), KTO (Ethayarajh et al., 2024), SimPO (Meng et al., 2024), and ODPO (Amini et al., 2024) further refine alignment with improved model efficiency and robustness. Most recently, S-DPO adapts alignment for user-item sequences, optimizing personalization by comparing with list-wise negative items. We provide a more detailed related work in Appendix C.

3 Preliminaries

We begin by formalizing the sequential recommendation task within the LM framework. Next, we outline a two-stage training paradigm that adapts existing LMs to the recommendation task, including *supervised fine-tuning (SFT)* and *preference alignment*. Centering around the alignment stage, we briefly introduce direct preference optimization (DPO) (Rafailov et al., 2024), a technique that aligns LMs using pairwise preference data; We then present S-DPO (Chen et al., 2024), a recent adaptation of DPO designed specifically for sequential recommendation.

Sequential Recommendation with LMs. Let $\mathcal{H}_u = [i^1, i^2, ..., i^{N_u}]$represent the chronologically ordered sequence of historical interactions for user u, where each element i^k encapsulates contextual details of the k-th interaction (e.g., item title, style, rating), and N_u denotes the total number of interactions. We define $\mathcal{H}_u^t = \mathcal{H}_u[:t]$ as as the subset of interactions up to time t, and let i_p^{t+} denote the ***next recent favorable (high-rated)*** item following the interaction history at t. Let π_θ be the LM performing the task, parameterized by θ. The sequential recommendation task within the LM framework is formulated as follows: given user u's interaction history $\mathcal{H}_u^t$ up to time t and a candidate item set $\mathcal{C} = \{i^{(j)}\}_{j=1}^K$, where $\mathcal{H}_u^t \cap \mathcal{C} = \emptyset$ and $i_p^{t+} \in \mathcal{C}$, the model π_θ is required to predict the item that most likely be favorable to user, i.e., i_p^{t+}.

Supervised Fine-tuning LMs for Sequential Recommendation. Supervised fine-tuning (Ouyang et al., 2022) (SFT) is widely adopted to adapt general-purpose LMs to recommendation tasks (Liao et al., 2024; Bao et al., 2023). Let $\mathbf{x}_u^t$ be the task prompt that encompasses user u's interaction history $\mathcal{H}_u^t$ up to time t, the candidate item set $\mathcal{C}$, and other task-related descriptions. We define $\mathcal{T}_c$ as the textual descriptions of candidate items in $\mathcal{C}$, and $\mathbf{y}_p^t$ as the text mapping of item $i_p^{t+} \in \mathcal{C}$ that best aligns with $\mathbf{x}_u^t$'s description. We construct the SFT training dataset $\mathcal{D}_{\text{SFT}}$ using pairwise data $(\mathbf{x}_u^t, \mathbf{y}_p^{t+}), \forall u, \forall t < N_u$, and frame the sequential recommendation as a sentence completion task. The objective that optimizes π_θ is:

$$\max_\theta \quad \mathbb{E}_{(\mathbf{x}_u^t, \mathbf{y}_p^{t+}) \sim \mathcal{D}_{\text{SFT}}} \left[\log \pi_\theta(\mathbf{y}_p^{t^+} | \mathbf{x}_u^t) \right]. \quad (1)$$

The LM fine-tuned with this objective on $\mathcal{D}_{\text{SFT}}$ is denoted as π_{SFT}. For brevity, we omit the timestamp signs in all subsequent equations unless its inclusion is essential for clarity.

Aligning LLM with Human Preference Feedback. While optimizing the SFT objective effectively adapts LMs to the downstream task, recent studies indicate that models still struggle to align outputs with human judgments of quality (Ziegler et al., 2019; Stiennon et al., 2020; Rafailov et al., 2024). To address this, a reward model $r(\mathbf{x}, \mathbf{y})$ is introduced to estimate output quality assessed by humans, aiming to maximize the expected reward.

To train the reward model, a dataset of comparisons $D = \{\mathbf{x}^{(i)}, \mathbf{y}_w^{(i)}, \mathbf{y}_l^{(i)}\}_{i=1}^N$ is constructed, where $\mathbf{y}_w^{(i)}$ and $\mathbf{y}_l^{(i)}$ denotes the preferred and dispreferred output generated based on $\mathbf{x}^{(i)}$, respectively. The alignment objective with the learned reward function is then defined as:

$$\max_\theta \quad \mathbb{E}_{x \sim \mathcal{D}, \mathbf{y} \sim \pi_\theta(\cdot|\mathbf{x})} \Big([r(\mathbf{x}, \mathbf{y})] - \beta D_{\text{KL}} \left[\pi_\theta(\mathbf{y}|\mathbf{x}) \| \pi_{\text{ref}}(\mathbf{y}|\mathbf{x}) \right] \Big), \quad (2)$$

where β is the parameter controlling the deviation from the reference model π_{ref}, and π_{SFT} is commonly used as the reference model. Based on Equation 2, a recent work DPO (Rafailov et al., 2024), employs the Bradley-Terry (Bradley and Terry, 1952) (BT), $P(\mathbf{y}_w \succ \mathbf{y}_l \mid \mathbf{x}) = \sigma(r(\mathbf{x}, \mathbf{y}_w) - r(\mathbf{x}, \mathbf{y}_l))$, to express the probability of human preference data in terms of the optimal policy rather than the reward model, they derive the objective based on pairwise preference data as:

$$\min_\theta - \mathbb{E}_{(\mathbf{x}, \mathbf{y}_w, \mathbf{y}_l) \sim D} \Big[\log \sigma \Big(\beta \log \frac{\pi_\theta(\mathbf{y}_w|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_w|\mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_l|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_l|\mathbf{x})} \Big) \Big]. \quad (3)$$

The above preference modeling paradigm aligns naturally with recommendation tasks, with both being preference-based decision-making. Building upon DPO, a recent effort named S-DPO (Chen et al., 2024) is proposed to further align LLM-based recommenders to user preference. They propose to pair each positive item with multiple negative items generated by random sampling as preference data, and revise the alignment objective as:

$$\min_\theta - \mathbb{E}_{(\mathbf{x}, \mathbf{y}_w, \mathcal{T}_d) \sim D} \Big[\log \sigma \Big(- \log \sum_{\mathbf{y}_d \in \mathcal{T}_d} \exp \Big(\beta \log \frac{\pi_\theta(\mathbf{y}_d|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_d|\mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_p|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_p|\mathbf{x})} \Big) \Big) \Big], \quad (4)$$

where $\mathcal{T}_d = \mathcal{T}_c \setminus \{\mathbf{y}_p\}$ contains the item titles of multiple dispreferred items*.

4 Does Comprehensive Preference Feedback Help?

We design a proof-of-concept experiment to assess how explicit preference granularity (via user-item ratings) impacts the next favorable item recommendation. We devise four-tier input configurations that progressively integrates preference signals: ***(i) Filtered Items***: Excludes negative-feedback items and provides no ratings, mimicking S-DPO's setup; ***(ii) Full Items***: Retains all historical items but still provides no ratings; ***(iii) Filtered Items + Rating***: Provides ratings but excludes negative-feedback items; ***(iv) Full Items + Rating***: Retains all items and their corresponding ratings. We fine-tune LLaMA3-8B on MovieLens and Amazon-Books (described Section 6.1) using the four input configurations. The experimental results are reported using Hit Ratio@1 (see Section 6.1, where higher values indicate better performance) and are shown in Figure 2. We observe that incorporating more fine-grained feedback in the form of ratings consistently improves performance. Although the objective is to predict probable favorable items, including negatively rated items with ratings in the user history proves beneficial, as aversion modeling is crucial for building a more accurate user profile. However, the *Full Items* configuration (without ratings) underperforms the *Filtered Items* configuration, as the absence of explicit annotations for negative items introduces noise into the learning process. These results highlight that explicit ratings help resolve ambiguity, enabling LLMs to differentiate between preferences and aversions.

5 Methodology

In this section, we first present how we leverage user historical interaction data to design the prompts, establishing the foundation for preference modeling in recommendations. We then introduce RecPO, a novel preference optimization framework for sequential recommendation that dynamically calibrates reward margins between preferred and dispreferred items according to their contextualized user-item relevance. The architectural workflow of our proposed framework is illustrated in Figure 3.

*We use positive/negative, as well as preferred/dispreferred interchangeably in the following content.

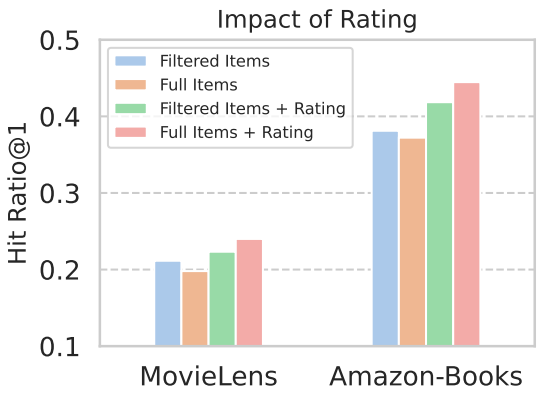


Figure 2: Impact of employing ratings and retaining low-rating items, where **"Filtered Items"** include no rating signals and remove all low-rating items from historical sequences, **"Full Items"** keep all historically interacted items and includes no rating signals,**"Filtered Items + Rating"** include rating signals but still remove low-rating items, **"Full Items + Rating"** retain both of them.

5.1 Complete Historical Interactions Enhance User Profiling

Sequential recommendation aim to predict subsequent preferred items for users based on their chronological interaction history. While existing approaches (Liao et al., 2024; Chen et al., 2024) remove items with negative feedback from interaction histories to construct homogeneous sequences (i.e., items with only positive feedback), in Section 4 we demonstrate that this practice can impair the fidelity of user preference profiling and induce performance degradation (see Figure 2), as it discards critical behavioral signals. In contrast, our framework preserves the complete interaction sequence for each user, explicitly retaining all historical items along with their associated preference feedback. We use user-item ratings to represent a hierarchical structure of preference feedback. Following prior work (Chen et al., 2024), the input prompts are composed of the following parts:

User historical interaction $\mathcal{H}_u$ Each item in the user history is formatted as `"[ItemTitle] | Rating: [ItemRating]"`. For example, `"Toy Story | Rating: 4"`. All historical items are concatenated with `"\n"` being the separator.

Candidate item set $\mathcal{C}$ We format all candidate items in a similar format as the historical items, except that no rating attributes are provided.

Task Description To facilitate LLM's understanding of user preference and the task, we prepend the history-specific prefixes (e.g., `"Given`

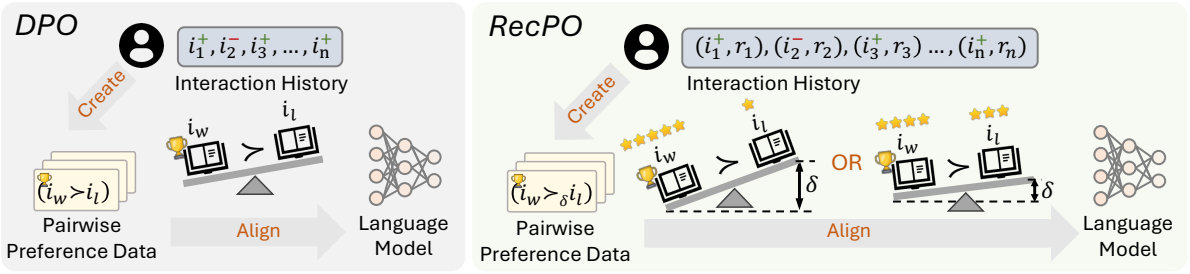


Figure 3: Illustrations for DPO and our framework: DPO assigns static preference margin across pairwise preference data, while our framework adaptively adjusts preference margin (δ) based on preference difference.

`the user's recent viewing and rating history`") and candidate-specific prefixes (e.g., "`recommend a movie they'll likely watch next and rate generously from following candidates`") to their respective sequences.

We concatenate the above components as the final textual input $\mathbf{x}_u$ to the LMs, with concrete examples for the constructed prompts in Appendix B. The ground truth item $\mathbf{y}_p$ and the negative items $\mathbf{y}_d \in \mathcal{T}_d$ are the titles of the respective items.

5.2 Adaptive Reward Margin Emulates Hierarchic Human Preference Difference

Current DPO-based methods, as introduced in Section 3, reduce preference modeling to maximizing the reward difference between pairwise preferred and dispreferred responses/items. This simplification exposes them to two key limitations in recommendation: (i) Neglecting preference hierarchy, where in reality, users may strongly prefer certain items while only slightly preferring others, compared to either the same or different negative items; (ii) Neglecting the time discounting effect, where users typically prioritize immediate satisfaction over delayed rewards. To incorporate both factors into model preference alignment, inspired by prior work (Meng et al., 2024; Amini et al., 2024), we propose an adaptive target reward margin term γ_r, which is dynamically determined by two key elements: the ratings of the two compared items and the time elapsed since their interactions relative to the current timestamp. Specifically, we utilize a utility function $\phi(\cdot)$ to evaluate the reward of an item—the higher the rating of an earlier interaction, the larger the utility. We define the margin of a pairwise data $(\mathbf{y}_p, \mathbf{y}_d)$ as follows:

$$\gamma_r = \lambda \frac{\phi\left(s_p, \Delta_{t_p}\right)}{\phi\left(s_d, \Delta_{t_d}\right)}, \tag{5}$$

where $\mathbf{y}_p$ is preferred over $\mathbf{y}_d$, λ controls the margin's magnitude, and $\Delta_{t_p} = t_p^+ - t$ indicates the time latency of the interaction. In this work, we set $\phi\left(s, \Delta_t\right) = s / \left(\Delta_t\right)^{0.5}$. Note that the choice of score function is customizable as long as it reflects the above preference rules. That is, $\phi\left(s, \Delta_t\right) \propto s / \left(\Delta_t\right)^{\alpha}$, where $\alpha > 0$ indicates the temporal decay factor. For dispreferred/negative items from either negative sampling or historical interactions, no user-assigned ratings are available and we set a default rating and time latency to facilitate the training. More details about the default value can be found in the Section 6.

5.3 Human Preference Alignment

We plug Equation 5 into the BT model to derive the distribution for pairwise preference data:

$$P^*(\mathbf{y}_p \succ \mathbf{y}_d \mid \mathbf{x}_u) = \sigma\left(r\left(\mathbf{x}_u, \mathbf{y}_p\right) - r\left(\mathbf{x}_u, \mathbf{y}_d\right) - \gamma_r\right) \tag{6}$$

In sequential recommendation, where each preferred item are paired with multiple dispreferred items, we leverage the Plackett-Luce (PL) model (Plackett, 1975; Luce, 1959) to generalize pairwise comparisons to a list-wise ranking framework. Formally, given the prompt x_u^t encompassing all the historical interactions of user u, a candidate set $\mathcal{C} = \{i_p\} \cup \mathcal{I}_d$ containing K items (one preferred item and $K-1$ dispreferred items), and a permutation σ representing the predicted ranking of these candidates based on user preference for the next item (denote $\sigma(j)$ as the item ranked at position j), the probability of observing the candidates' preference ranked as $[\mathbf{y}_{\sigma(1)}, \mathbf{y}_{\sigma(2)}, \ldots, \mathbf{y}_{\sigma(K)}]$ is:

$$P(\sigma \mid \mathbf{x}_u, \mathcal{T}_c) = \prod_{j=1}^{K} \frac{\exp\left(r\left(\mathbf{x}_u, \mathbf{y}_{\sigma(j)}\right)\right)}{\sum_{m=j}^{K} \exp\left(r\left(\mathbf{x}_u, \mathbf{y}_{\sigma(m)}\right)\right)}. \tag{7}$$

Finally, we combine Equations 6 and 7 into the final objective shown in Equation 8. Note that our method is reduced to S-DPO when $\lambda = 0$. For brevity, the detailed derivation process is provided in Appendix A. By optimizing the derived objective, we effectively integrate explicit rating signals

$$\mathcal{L}\left(\pi_\theta; \pi_{\text{ref}}\right) = -\mathbb{E}_{(\mathbf{x}_u, \mathbf{y}_p, \mathcal{T}_d)\sim\mathcal{D}}\left[\log\sigma\left(-\log\sum_{\mathbf{y}_d\in\mathcal{T}_d}\exp\left(\beta\log\frac{\pi_\theta(\mathbf{y}_d \mid \mathbf{x}_u)}{\pi_{\text{ref}}(\mathbf{y}_d \mid \mathbf{x}_u)}\right.\right.\right.$$
$$\left.\left.\left. - \beta\log\frac{\pi_\theta(\mathbf{y}_p \mid \mathbf{x}_u)}{\pi_{\text{ref}}(\mathbf{y}_p \mid \mathbf{x}_u)} - \lambda\frac{\phi(s_p, \Delta_{t_p})}{\phi(s_d, \Delta_{t_d})}\right)\right)\right]. \tag{8}$$

with temporal decay factors to refine the implicit reward mechanism, enabling LLM recommenders to better learn user preference patterns in real-world recommendation scenarios.

6 Experiment

6.1 Setup

Datasets. We select four publicly available representative recommendation benchmark datasets for our experiments: *(1) MovieLens-1M*[†], sourced from the MovieLens platform, contains 1 million ratings from 6,000 users on 4,000 movies; *(2) Amazon-books*[‡] is a subset of the Amazon Review dataset containing 22 million user interactions, reviews, and ratings for 2 million books from 8 million users; *(3) Steam*[§] is a dataset of user interactions with games, including game purchases, playtime, and reviews from the Steam platform; *(4) BeerAdvocate*[¶] collects beer reviews covering multiple sensory aspects and overall ratings.

For each dataset, we apply k-core filtering (He and McAuley, 2016) to remove users and items with less than $k = 5$ interactions. We then construct a candidate set of 20 items from which the model selects. During training, this set is composed of 10 subsequent interactions (ensuring that the correct item is always included) and 10 randomly sampled non-interacted items. For validation and testing, the candidate set consists of the correct item plus 19 randomly sampled non-interacted items. For ML-1M, Amazon-books, and BeerAdvocate, we utilize rating feedback to adjust the preference margin, and for Steam where explicit ratings are unavailable, we instead rely on play-hours as a proxy for user preferences. For each user, we order the interactions chronologically, using the second-last interaction for validation, the last one for testing, and the rest for training. More details about the datasets we use are provided in Appendix D.1.

[†] https://grouplens.org/datasets/movielens/1m/

[‡] https://nijianmo.github.io/amazon/index.html

[§] https://github.com/kang205/SASRec

[¶] https://cseweb.ucsd.edu/~jmcauley/datasets.html#multi_aspect

Baselines. We compare RecPO with two types of baseline models: (i) *Traditional* methods leverage sequential patterns in user behaviors to predict the next interacted item, using various modeling architectures such as recurrent neural networks (GRU4Rec (Hidasi, 2016)), convolutional neural networks (Caser (Tang and Wang, 2018)), or multi-head self-attention frameworks (SASRec (Kang and McAuley, 2018)). (ii) *LM-based* methods utilize LMs to process historical interactions and predict the next interacted item. In addition to evaluating the general-purpose LM, LLaMA3 (Dubey et al., 2024) and the standard preference optimization baseline DPO (Rafailov et al., 2024), we include SimPO (Meng et al., 2024), a reference-free method that enhances DPO with length regularization and fixed margin term, and S-DPO (Chen et al., 2024), which adapts DPO for specically for sequential recommendation. More details about the baselines are provided in Appendix D.2.

Implementation. All experiments were performed on no more than 8 NVIDIA RTX A6000 with 48GB of VRAM. For our method and all the other preference learning approaches, we first conduct SFT adapt them to the recommendation task. Then in alignment, models were initialized from SFT checkpoints and optimized using the alignment loss defined in Equation 8. More implementation details are provided in Appendix D.3.

Evaluation Metrics. We follow S-DPO to evaluate the models using two metrics: Hit Ratio@1, which quantifies recommendation accuracy as the proportion of test cases where the model's top-ranked prediction matches the ground-truth next interacted item, and Valid Ratio, which measures instruction compliance by calculating the proportion of recommendations that adhere to formatting guidelines and belong to the predefined candidate set. The latter ensures the model avoids generating out-of-distribution or irrelevant items. Together, these metrics holistically assess both the precision of recommendations and their alignment with practical deployment constraints.

	Method	MovieLens		Amzon-Books		Steam		BeerAdvocate	
		HR@1	ValidRatio	HR@1	ValidRatio	HR@1	ValidRatio	HR@1	ValidRatio
Tranditional	GRU4Rec	0.2664	1.0000	0.1310	1.0000	0.4584	1.0000	0.3708	1.0000
	Caser	0.2714	1.0000	0.1538	1.0000	0.4394	1.0000	0.3757	1.0000
	SASRec	0.2671	1.0000	0.1559	1.0000	0.4587	1.0000	0.3800	1.0000
LLM-based	LLaMA3	0.0929	0.7351	0.0654	0.6165	0.0852	0.8672	0.0686	0.6617
	SFT	0.2478	0.9985	0.4447	0.9974	0.3122	0.9990	0.2645	0.9936
	DPO	0.2809	0.9970	0.5049	0.9887	0.3340	0.9980	0.4412	0.9875
	SimPO	0.2974	0.9725	0.5129	0.9564	0.3401	0.9766	0.4020	0.9250
	S-DPO	0.2902	0.9983	0.5065	0.9880	0.3588	0.9990	0.4698	0.9903
Ours	RecPO	**0.3451**	0.9969	**0.5802**	0.9851	**0.4672**	0.9985	**0.5771**	0.9887

Table 1: Overall model performance comparison on four real-world recommendation datasets. Hit Ratio@1 and Valid Ratio are reported, the best performance is bolded and runner-ups are underlined.

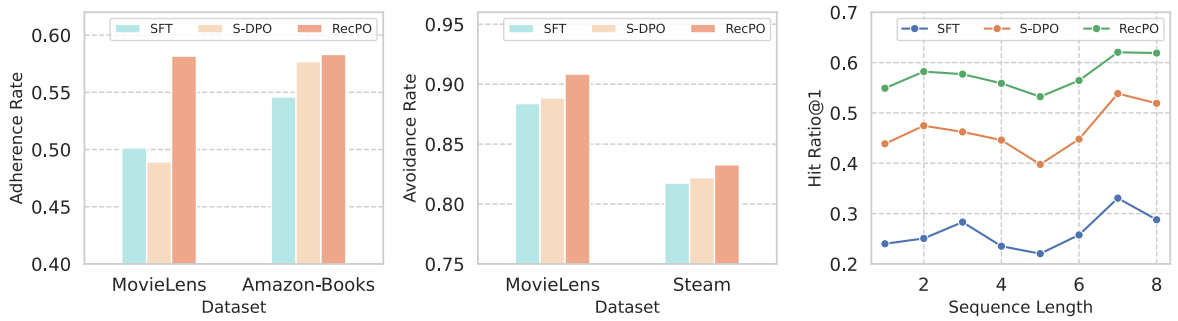
6.2 Main Results and Ablations

Overall Performance. Table 1 compares RecPO with the baselines across four sequential recommendation datasets, revealing three key findings: ***(i) SFT bridges the gap between LLMs and recommendation constraints.*** While LLMs (e.g., LLaMA 3) possess open-world knowledge, their raw outputs often violate practical requirements (e.g., recommending non-candidate items or exceeding item limits). SFT significantly improves valid output rates, achieving parity with traditional recommenders, demonstrating the necessity to align general-purpose LLMs with structural and behavioral requirements of real-world recommendation applications; ***(ii) Preference optimization further unlocks the potential of LLMs in recommendation.*** All preference learning methods, including our proposed RecPO, DPO, SimPO, and S-DPO, significantly outperform SFT in Hit Ratio@1, suggesting the alignment between preference optimization and ranking-centric recommendation objectives. Notably, RecPO and S-DPO surpass the standard DPO, demonstrating that multi-negative preference learning better captures nuanced user preference patterns in recommendation scenarios. Although SimPO achieves impressive improvement in Hit Ratio@1, it exhibits a noticeable degradation in Valid Ratio compared to other approaches, which highlights the limitations of reference-free optimization in mitigating distributional discrepancies between recommendation tasks and general NLP tasks; ***(iii) RecPO achieves SOTA results across all benchmarks.*** By integrating explicit ratings with adaptive reward margins, RecPO improves Hit Ratio@1 by 13.12% to 30.21% over other LLM-based approaches. We attribute this to its human-aligned preference modeling grounded in cognitive science principles. While RecPO also outperforms all the traditional recommenders, its performance gain on the Steam dataset is relatively small. We posit that the narrower performance gap stems from the play-hour-derived ratings, demonstrating homogeneous interaction patterns that traditional models with simple structures can still capture.

Dataset	Log Diff	Log Ratio	RecPO
MovieLens	0.3160	0.3247	0.3451
Amazon-Books	0.5370	0.5455	0.5802
Steam	0.4284	0.4517	0.4672
BeerAdvocate	0.5023	0.5257	0.5771

Table 2: Ablation study on the margin term function, Hit Ratio@1 is reported for comparison.

Ablation Study on Margin Functions. We denote ϕ_p and ϕ_d as the scores for the preferred and dispreferred items respectively for brevity. By default, RecPO defines the margin term γ_t as the ratio of preference scores ϕ between positive and negative item pairs, as formalized in Equation 5. To evaluate the impact of this design choice, we introduce two alternative margin functions: *(i) Log Diff*, $\gamma_r = \lambda \log(\phi_p - \phi_d)$; *(ii) Log Ratio*, $\gamma_r = \lambda(\log \phi_p - \log \phi_d)$. As shown in Table **??**, both variants outperform the strongest LLM-based recommender baseline, confirming the general utility of margin-aware optimization. However, RecPO 's default margin formulation achieves superior performance across all datasets. This advantage arises because the default ratio-based margin amplifies gradients during training, particularly when historical user ratings exhibit low volatility. By directly contrasting ϕ_p and ϕ_d through division, the model receives stronger learning signals to prioritize subtle yet critical preference patterns.



(a) Contextualized Preference Adherence (b) Unfavorable Item Avoidance (c) Sequence Length Robustness

Figure 4: Comparative analysis of SFT, S-DPO, and RecPO: (a) Context-aware user preference adherence, (b) Unfavorable item recommendation avoidance, and (c) Robustness across varying user history lengths.

6.3 In-depth Preference Alignment Analysis

RecPO adheres to contextualized preferences. To assess RecPO 's ability to leverage contextualized user preferences for next-item recommendations, we construct test sets from MovieLens and Amazon-Books in which candidate sets include multiple high-rated items from users' subsequent interactions, and follow the rules in Section 3 to distinguish positive and negative candidates. Specifically, we use ***Adherence Rate*** (see Appendix D.4) to measure the model effectiveness in adhering to contextualized user preferences by recommending the immediate high-rated item for the next interaction. As shown in Figure 4(a), RecPO consistently outperforms SFT and S-DPO in ranking the immediately next high-rated item above temporally distant ones, demonstrating its enhanced capacity to model contextualized user intent and deliver timely recommendations. In contrast, S-DPO fails to consistently outperform SFT, indicating deviations from true preference hierarchy. These results suggest that RecPO 's adaptive reward margins enable recommendations closely adhere to human preference hierarchy.

RecPO avoids undesirable recommendations. Beyond modeling contextualized user preferences, we further evaluate the model's ability to avoid recommending low-rated items. For this analysis, we construct test sets from the MovieLens and Steam datasets by augmenting candidate sets with low-rated items from users' subsequent interactions. Similarly, we use ***Avoidance Rate*** (see Appendix D.4) to assess the model's effectiveness in avoiding the recommendations of unfavorable (unsatisfactory) items for the next interaction. As shown in Figure 4(b), RecPO consistently achieves the highest low-rated item avoidance rates across benchmarks, outperforming all baselines. These results suggest that incorporating explicit ratings for more comprehensive preference feedback simultaneously captures patterns for both desirable and undesirable items. As a result, RecPO minimizes the risk of recommending irrelevant or disliked items, a crucial factor in sustaining user engagement.

RecPO is robust to variations in user sequence length. In Figure 4(c), we investigate RecPO 's robustness to variations in historical interaction lengths using the BeerAdvocate dataset. For this analysis, we partition the test set into subsets categorized by the length of historical interactions and evaluate the performance per subset. RecPO exhibits sustained efficacy, delivering larger performance margins over SFT than S-DPO across all length groups. While all three methods follow analogous performance trajectories as history length increases, RecPO maintains the most stable results, evidenced by the lowest variance in Hit Ratio@1 scores (8.7% vs. 17.8% for S-DPO). This observation shows RecPO 's superior adaptability to varying context lengths—a critical trait for real-world systems with inherently dynamic user interactions.

7 Conclusion

In this paper, we investigate the problem of aligning recommendation objectives with human cognitive processes by proposing RecPO, a novel preference optimization algorithm designed to instill nuanced user preference patterns into LLM-based recommenders. Our approach leverages more comprehensive preference feedback, and features an adaptive reward margin mechanism that dynamically calibrates the relative preference between positive and negative item pairs, leveraging explicit rating signals and interaction latency to capture fine-grained, context-aware user preferences. Extensive experiments across multiple benchmarks

demonstrate that RecPO outperforms state-of-the-art methods, achieving performance improvements ranging from 13.12% to 30.21%, while exhibiting superior adherence to contextualized preferences and a reduced retrieval rate for unfavorable items.

Limitations

While we demonstrate that incorporating more comprehensive interaction feedback improves user profiling, this work focuses solely on ratings as the key factor for modeling preference hierarchy. In real-world platforms, other user interaction signals (e.g., clicks, reviews, session duration) can provide additional insights into user cognitive behaviors and enhance preference-based decision-making. Although our results show improved recommendation performance and alignment with human perspectives, further studies are needed to evaluate the model's user profiling capabilities beyond a single metric. Emulating human cognitive behaviors should extend beyond predicting the most likely interacted item toward modeling broader, more complex user behavior within the system.

References

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Direct preference optimization with an offset. In *Findings of the ACL*.

Janet Wilde Astington and Jennifer M Jenkins. 1995. Theory of mind development and social understanding. *Cognition & Emotion*.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*.

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 *RecSys*.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*.

Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *SIGIR*.

Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent contrastive learning for sequential recommendation. In *WWW*.

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

James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The youtube video recommendation system. In *RecSys*.

Yashar Deldjoo, Markus Schedl, Paolo Cremonesi, and Gabriella Pasi. 2020. Recommender systems leveraging multimedia content. *ACM Computing Surveys (CSUR)*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *ACM Web Conference*.

Xinyan Fan, Zheng Liu, Jianxun Lian, Wayne Xin Zhao, Xing Xie, and Ji-Rong Wen. 2021. Lighter and better: low-rank decomposed self-attention networks for next-item recommendation. In *SIGIR*.

Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524*.

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 *RecSys*.

Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging large language models for sequential recommendation. In *RecSys*.

Ruining He and Julian McAuley. 2016. Vbpr: visual bayesian personalized ranking from implicit feedback. In *AAAI*.

B Hidasi. 2016. Session-based recommendations with recurrent neural networks. In *ICLR*.

Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *ACM Recommender Systems conference*.

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

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text is all you need: Learning language representations for sequential recommendation. In *SIGKDD*.

Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *WSDM*.

Yaoyiran Li, Xiang Zhai, Moustafa Alzantot, Keyi Yu, Ivan Vulić, Anna Korhonen, and Mohamed Hammad. 2024. Calrec: Contrastive alignment of generative llms for sequential recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 422–432.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-recommendation assistant. In *SIGIR*.

R Duncan Luce. 1959. *Individual choice behavior*, volume 4. Wiley New York.

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

Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *ACM International Conference on Information and Knowledge Management*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. SimPO: Simple preference optimization with a reference-free reward. In *NeurIPS*.

Long Ouyang, Jeffrey Wu, Xu Jiang, et al. 2022. Training language models to follow instructions with human feedback. *NeurIPS*.

Robin L Plackett. 1975. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2000. Analysis of recommendation algorithms for e-commerce. In *ACM Conference on Electronic Commerce*.

J Ben Schafer, Joseph Konstan, and John Riedl. 1999. Recommender systems in e-commerce. In *ACM conference on Electronic commerce*.

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *NeurIPS*.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*.

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

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *ICDE*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*.

Shenghao Yang, Weizhi Ma, Peijie Sun, Qingyao Ai, Yiqun Liu, Mingchen Cai, and Min Zhang. 2024. Sequential recommendation with latent relations based on large language model. In *SIGIR*.

Feng Yu, Yanqiao Zhu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2020. Tagnn: Target attentive graph neural networks for session-based recommendation. In *SIGIR*.

Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id-vs. modality-based recommender models revisited. In *SIGIR*.

Junjie Zhang, Ruobing Xie, Yupeng Hou, Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. *ACM Transactions on Information Systems*.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *CIKM*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Derivation of Preference Distribution

In the standard Bradley-Terry model, the probability that candidate i beats candidates j is

$$P(\mathbf{y}_i \succ \mathbf{y}_j) = \sigma\left(r\left(\mathbf{x}, \mathbf{y}_w\right) - r\left(\mathbf{x}, \mathbf{y}_l\right)\right) = \frac{\exp\left(r\left(\mathbf{x}_u, \mathbf{y}_i\right)\right)}{\exp\left(r\left(\mathbf{x}_u, \mathbf{y}_i\right)\right) + \exp\left(r\left(\mathbf{x}_u, \mathbf{y}_i\right)\right)}, \tag{9}$$

We will only use w_i to represent the candidate-specific probability $\exp\left(r\left(\mathbf{x}_u, \mathbf{y}_i\right)\right)$ in subsequent equations for brevity. Now suppose we wish to include a margin term γ_{ij}, then we can define the pairwise probability as

$$P(\mathbf{y}_i \succ \mathbf{y}_j) = \frac{w_i \exp\left(-\gamma_{ij}\right)}{w_i \exp\left(-\gamma_{ij}\right) + w_j} \tag{10}$$

where we assume $\gamma_{ij} = -\gamma_{ji}$. Specifically, we can use the Plackett-Luce model decomposes a ranking $i_1 \succ i_2 \succ i_k \succ \cdots \succ i_K$ into sequential choices competition. Therefore, at each step t, the wining (got selected) probability i_k is proportional to its weight, i.e., $w_k = \exp\left(r\left(\mathbf{x}_u, \mathbf{y}_k\right)\right)$. Now the added margin term γ_{ij} modifies the competition by giving each candidate an extra boost (or penalty) when facing an opponent. In other words, when candidate i competes against candidate j (within the remaining set) its effective strength is boosted by the factor $\exp(-\gamma_{ij})$. Then, by an extension of Luce's choice axiom, we can get the probability of choosing candidate i from the set $\mathcal{C}$ is proportional to its effective weight:

$$P(i \text{ chosen from } \mathcal{C}) = \frac{w_i \exp\left(-\sum_{j \in \mathcal{C} \setminus \{i\}} \gamma_{ij}\right)}{\sum_{k \in \mathcal{C}} w_k \exp\left(-\sum_{j \in \mathcal{C} \setminus \{k\}} \gamma_{kj}\right)}. \tag{11}$$

Let $\sigma = (\sigma(1), \sigma(2), \ldots, \sigma(K))$ be a full ranking of K candidates. We construct the ranking sequentially. At step r, let

$$\mathcal{C}_r = \mathcal{C} \setminus \{\sigma(1), \sigma(2), \ldots, \sigma(r-1)\} \tag{12}$$

be the remaining set. Then the probability that candidate $\sigma(r)$ is selected at step r will be,

$$P(\sigma(r) \mid \sigma(1), \ldots, \sigma(r-1)) = \frac{w_{\sigma_{(r)}} \exp\left(-\sum_{j \in \mathcal{C}_r \setminus \{\sigma(r)\}} \gamma_{\sigma(r)j}\right)}{\sum_{k \in \mathcal{C}_r} w_{\sigma_{(k)}} \exp\left(-\sum_{j \in \mathcal{C}_r \setminus \{k\}} \gamma_{kj}\right)}. \tag{13}$$

We can thereby get the likelihood of the full ranking by the chain rule,

$$P(\sigma \mid \mathcal{C}) = \prod_{r=1}^{K-1} \frac{w_{\sigma_{(r)}} \exp\left(-\sum_{j \in \mathcal{C}_r \setminus \{\sigma(r)\}} \gamma_{\sigma(r)j}\right)}{\sum_{k \in \mathcal{C}_r} w_{\sigma_{(k)}} \exp\left(-\sum_{j \in \mathcal{C}_r \setminus \{k\}} \gamma_{kj}\right)} \tag{14}$$

In the recommendation setting we are especially interested in penalizing the positive item's "win" relative to each negative, which means one might only apply a margin from the positive item to each negative. Therefore, we can derive the preference distribution of recommendation case given interactions $\mathbf{x}_u$ of user u, multiple negative items $\mathbf{y}_d \in \mathcal{T}_d$ and the positive item $\mathbf{y}_p$:

$$P(\mathbf{y}_p \succ \mathbf{y}_d, \forall \mathbf{y}_d \in \mathcal{T}_d \mid \mathbf{x}_u, \mathbf{y}_p, \mathcal{T}_d) = \frac{w_p \exp\left(-\sum_{j=1}^{K-1} \gamma_{p,d_j}\right)}{w_p \exp\left(-\sum_{j=1}^{K-1} \gamma_{p,d_j}\right) + \sum_{j=1}^{K-1} w_{d_j}}. \tag{15}$$

Notably, the ranking likelihood would reduce to the standard Plackett–Luce model if the margin term $\gamma = 0$ for all pairs.

B Prompt Examples

We refer to the prompts used in previous works (Chen et al., 2024; Liao et al., 2024) to build the prompt examples in Figure 5 for recommendation preference data generation.

C Related Work

Sequential Recommendation. Sequential recommendation aims to model user preferences by capturing temporal patterns in interaction sequences. Early approaches, such as GRU4Rec (Hidasi, 2016), leveraged recurrent neural networks (RNNs) to encode sequential dependencies, while SASRec (Kang and McAuley, 2018) introduced self-attention mechanisms to better capture long-range dependencies. Convolutional-based methods like Caser (Chang et al., 2021) explored local patterns in sequences using convolutional filters. Recent state-of-the-art methods have further advanced the field by incorporating graph-based structures (Yu et al., 2020), contrastive learning (Xie et al., 2022; Chen et al., 2022), and hybrid architectures (Li et al., 2020; Zhou et al., 2020; Fan et al., 2021) for improved accuracy and robustness.

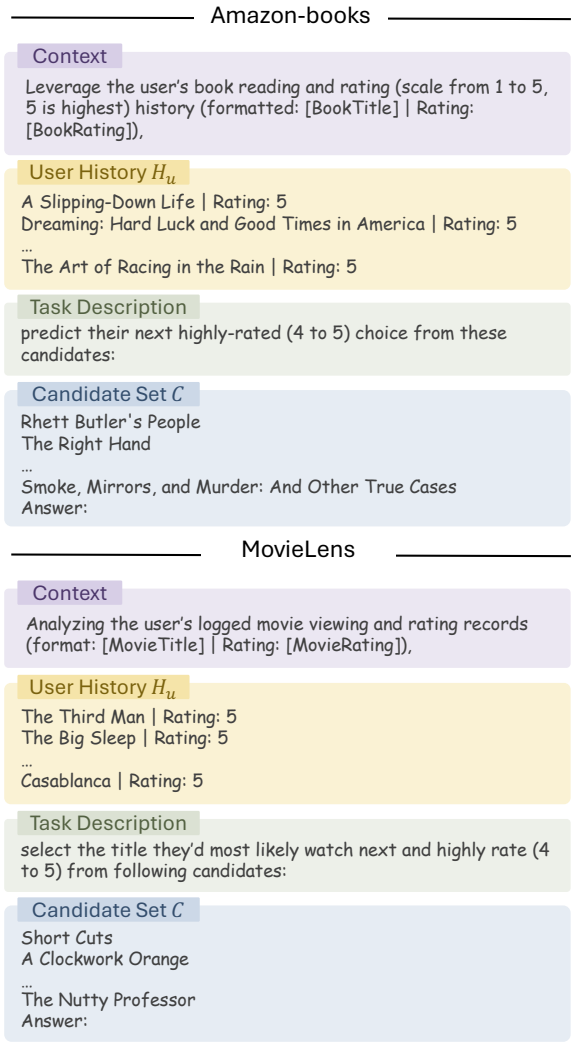


Figure 5: Textual prompt examples for Amazon-books and MovieLens.

LLMs for Recommendation. The integration of LLMs into sequential recommendation has gained momentum due to their ability to leverage rich semantic knowledge and contextual understanding. LLMs are typically integrated by encoding item descriptions, user reviews, or interaction histories as textual inputs, enabling the model to capture nuanced item characteristics and user preferences. For instance, LLaRA (Liao et al., 2024) employs classical sequential recommender systems to generate item embeddings, which are then fused with sequential interaction data to improve recommendation accuracy. TALLRec (Bao et al., 2023) fine-tunes LLMs on user-item interaction sequences, treating recommendations as a text generation task to predict the next item. Other approaches tackles the task from prompting (Geng et al., 2022; Gao et al., 2023; Lyu et al., 2023) or multi-modal data exploitation (Yuan et al., 2023). These methods demonstrate the potential of LLMs to bridge the gap between natural language understanding and sequential recommendation, enabling more interpretable and context-aware recommendations.

Dataset	# Sequence	# Items	# Interactions
MovieLens	6,040	3,952	994,169
Amazon-Books	5,103	38,203	62,290
Steam	3,171	4,251	82,072
BeerAdvocate	4,724	6,105	91,207

Table 3: Statistics of datasets

LLM Alignment. LLM alignment techniques aim to align general-purpose LMs' outputs with human preferences, ensuring that generated content is both useful and safe. While not specifically designed for recommendation tasks, these methods have inspired advancements in preference modeling. Early approaches like Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) and Proximal Policy Optimization (Schulman et al., 2017) laid the foundation by using reinforcement learning to fine-tune models based on human feedback. DPO (Rafailov et al., 2024) emerged as a simpler and more efficient alternative, directly optimizing preference data without requiring explicit reward modeling. Building on DPO, methods like IPO (Azar et al., 2024), CPO (Xu et al., 2024), KTO (Ethayarajh et al., 2024), SimPO (Meng et al., 2024), and ODPO (Amini et al., 2024) further refine alignment by addressing limitations such as capturing fine-grained preference hierarchies, reducing reward hacking, improving robustness to noisy feedback, and enhancing generalization across diverse user contexts. Most recently, S-DPO (Chen et al., 2024) adapts alignment techniques specifically for recommendation tasks, focusing on sequential user preferences and improving the personalization of LLM-based recommenders.

D Experimental Settings

D.1 Datasets

We use four widely used real-world sequential recommendation datasets for evaluation, including *MovieLens-1M*[‖], *Amazon-books*[**], *Steam*[††] and *BeerAdvocate*[‡‡]. We demonstrate the dataset statistics in Table 3. The MovieLens-1M dataset is sourced from the MovieLens platform and con-

[‖] https://grouplens.org/datasets/movielens/1m/
[**] https://nijianmo.github.io/amazon/index.html
[††] https://github.com/kang205/SASRec
[‡‡] https://cseweb.ucsd.edu/~jmcauley/datasets.html#multi_aspect

tains 1 million ratings from 6,000 users on 4,000 movies. The Amazon-Books dataset is a subset of the Amazon Review dataset and comprises 22 million user interactions, reviews, and ratings for 2 million books from 8 million users. The Steam dataset includes user interactions with games—such as purchases, playtime, and reviews—from the Steam platform, while the BeerAdvocate dataset collects beer reviews that cover multiple sensory aspects along with overall ratings. For each dataset, we filter out items and users with fewer than 20 interactions. To prevent information leakage during training and evaluation, we adopt the leave-last-two splitting method to divide the datasets into training, validation, and test sets. We build a candidate set of 20 items for each user sequence, from which the model selects the next item. During training, this set comprises 10 subsequent interactions (ensuring that the correct item is always included) and 10 randomly sampled non-interacted items. For validation and testing, the candidate set consists of the correct item plus 19 randomly sampled non-interacted items. To align with the task objective of recommending the most likely favorable item as the next interaction, we follow classical sequential recommendation settings by considering only highly rated items (ratings 4 to 5 on a scale of 1 to 5) from subsequent interactions as the positive item (i.e., the correct answer) (Li et al., 2024). The same process is applied to the validation and test sets; we only retain user sequences whose next item is highly rated. Meanwhile, we preserve all historical interactions and their corresponding ratings in the user history sequence for comprehensive user profiling. Since the Steam dataset lacks explicit rating signals, we use user play-hours as an implicit rating and convert it to a 1-to-5 scale based on its percentile ranking. For example, if a user's playtime for a game falls within the top 20% compared to other players, the corresponding user-item pair is assigned a rating of 5.

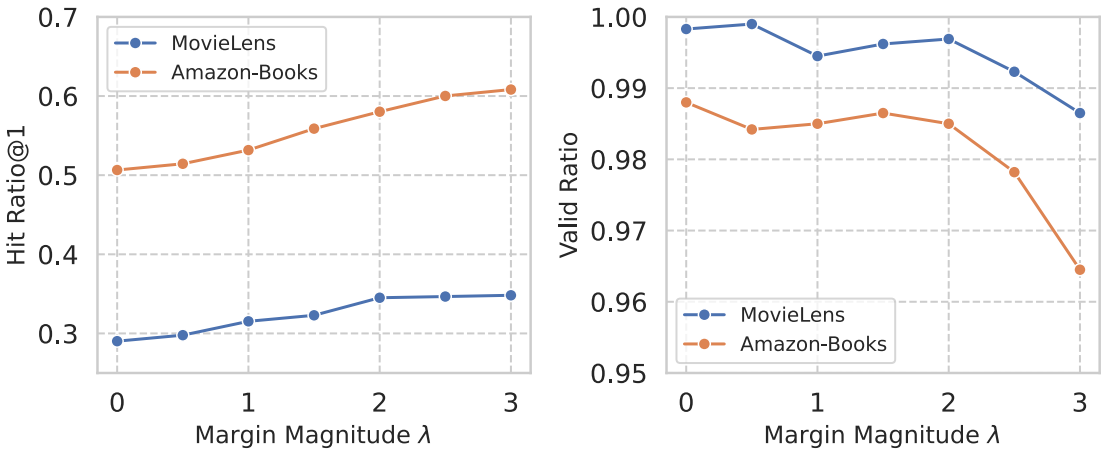
D.2 Baselines

We include the following baseline models for performance comparison:

- GRU4Rec (Hidasi, 2016) is a recurrent neural network-based model that captures sequential patterns in user interaction sequences session-based recommendation.
- Caser (Tang and Wang, 2018) is a convolutional neural network-based model that learns both local and sequential patterns in user-item interactions using convolutional filters.
- SASRec (Kang and McAuley, 2018)is a transformer-based model that leverages self-attention to capture long-range dependencies and dynamic user preferences in sequential recommendation.
- LLaMA-3 (Dubey et al., 2024) is a general-purpose LLM with strong semantic reasoning capabilities. We adapt it to sequential recommendation by treating it as a text prediction problem.
- DPO (Rafailov et al., 2024) is a preference alignment technique that fine-tunes models using pairwise preference data. In this work, we construct preference data based on explicit preference feedback.
- SimPO (Meng et al., 2024) is an extension of DPO that directly optimizes pairwise preferences without requiring explicit reward models or complex sampling strategies for improved efficiency and scalability.
- S-DPO (Chen et al., 2024) is a variant of DPO specifically adapted for sequential recommendation that incorporates list-wise negative items in preference alignment.

D.3 Implementation Details

All experiments were conducted on a maximum of 8 NVIDIA RTX A6000 GPUs, each with 48GB of VRAM. Our framework is implemented using Python 3.10.6, PyTorch 2.2.2, and Huggingface Transformers 4.43.3. For all LLM-based recommenders, we employ LLaMA 3.1 8B (Dubey et al., 2024) as the base model for both SFT and alignment. During training, we set the learning rate to 1e-5 for all LLM-based recommenders and use the AdamW optimizer. Additionally, we apply a 5% warm-up strategy and adjust the learning rate using a cosine scheduler. A global batch size of 128 is used to balance training efficiency and memory consumption. The maximum sequence length is tailored to each dataset based on the features involved and the average title lengths. We set $\beta = 1$ for all preference optimization approaches. For multi-negative preference learning, including S-DPO and our proposed RecPO, we adopt the S-DPO settings and fix the number of negatives at 3. In particular,



(a) Impact of λ on Hit Ratio@1 (b) Impact of λ on Valid Ratio

Figure 6: Sensitivity analysis of the margin parameter λ on recommendation performance: (a) Hit Ratio@1 and (b) Valid Ratio across MovieLens and Amazon-Books datasets.

we set the margin term in SimPO as 2 and set the parameter λ in our method as 2. Finally, following the prompt format provided in Appendix B, we create several additional prompt templates and randomly sample one for each user sequence during training and evaluation to ensure model flexibility and generality. For all traditional recommenders, we follow the settings from previous work (Chen et al., 2024) by setting the learning rate to 0.001, the batch size to 256, and using the Adam optimizer for model optimization.

D.4 Evaluation Metrics

As mentioned in Section 6.1, we primarily employ two metrics to evaluate model effectiveness: Hit Ratio@1, which measures how accurately the model recommends the correct item, and Valid Ratio, which assesses whether the model follows instructions to generate outputs in the required format. In Section 6.3, we introduce two additional metrics—***Adherence Rate*** and ***Avoidance Rate***—both derived from Hit Ratio@1. These metrics evaluate the model's ability to adhere to contextualized user preferences and avoid recommending unfavorable (unsatisfactory) items for the next interaction, with higher values indicating better performance. In our main experiment, the candidate sets during testing include the last item from the user's full sequence—typically a highly rated item (rating 4 to 5 on a scale of 1 to 5)—with the remaining candidates randomly sampled from the non-interacted set. In the contextualized preference adherence experiment, the candidate set for testing includes at least two high-rated items from the subsequent sequence. We follow the rule described in Section 3 to designate the positive item as the one with the smallest time latency Δ_t relative to the prediction timestamp t. A high ***Adherence Rate*** indicates that the model consistently recommends the positive item among all high-rated candidates. For the unfavorable item avoidance experiment, we construct the test set by selecting user sequences where the last interaction is low-rated (rating 1 to 2). Instead of measuring whether the model recommends this low-rated item, we assess whether it favors the randomly sampled candidates over the unfavorable item. Thus, a high ***Avoidance Rate*** signifies that the model successfully avoids recommending unfavorable items to users.

E Analysis on Margin Magnitude

As detailed in Section 5, the parameter λ controls the influence extent of the margin term γ_r on preference learning. We adopt $\lambda = 2$ as the default value to balance Hit Ratio@1 (recommendation accuracy) and Valid Ratio (instruction-following capability). To further study the impact of λ on model effectiveness, we conduct sensitivity analyses on MovieLens and Amazon-Books, with results visualized in Figure 6. Increasing λ consistently elevates Hit Ratio@1, though the rate of improvement diminishes at higher values (e.g., $\lambda = 3$). However, excessively large λ values degrade the Valid Ratio, which quantifies the model's adherence to user instructions. While Hit Ratio@1 reflects recommendation accuracy, maintaining a robust Valid Ratio ensures alignment with user intent. We recommend $\lambda \approx 2$ to harmonize both metrics.