What Makes LLMs Effective Sequential Recommenders? A Study on Preference Intensity and Temporal Context

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

Abstract

Sequential recommendation systems aspire to profile users by interpreting their interaction histories, echoing how humans make decisions by weighing experience, relative preference strength, and situational relevance. Yet, existing large language model (LLM)-based recommenders often fall short of mimicking the flexible, context-aware decision strategies humans exhibit, neglecting the structured, dynamic, and context-aware mechanisms fundamental to human behaviors. To bridge this gap, we propose RecPO, a preference optimization framework that models structured feedback and contextual delay to emulate human-like prioritization in sequential recommendation. RecPO exploits adaptive reward margins based on inferred preference hierarchies and temporal signals, enabling the model to favor immediately relevant items and to distinguish between varying degrees of preference and aversion. Extensive experiments across five real-world datasets demonstrate that RecPO not only yields performance gains over state-of-the-art baselines, but also mirrors key characteristics of human decision-making: favoring timely satisfaction, maintaining coherent preferences, and exercising discernment under shifting contexts. Code: https://anonymous.4open.science/r/RecPO-020A/

1 Introduction

Large language models (LLMs) are increasingly being adapted for sequential recommendation (Harte et al., 2023; Li et al., 2023; Yang et al., 2024; Bao et al., 2023; Zhang et al., 2023), where the task is to predict the next item a user will interact with based on their historical behaviors. Unlike traditional recommenders (Hidasi, 2016; Kang and McAuley, 2018; Tang and Wang, 2018), LLM-based systems leverage semantic understanding and reasoning capabilities to model user preferences from textual interaction histories.

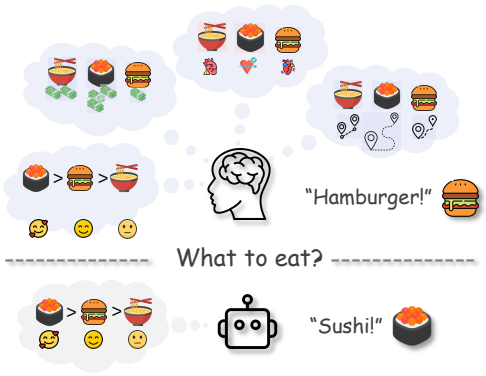


Figure 1: Human behaviors involve trade-offs among *preference intensity, satisfaction delay, effort, and risk*—factors largely overlooked in current LLM-based preference modeling.

Current approaches predominantly rely on preference alignment techniques such as DPO (Rafailov et al., 2024) and its variants (Chen et al., 2024; Meng et al., 2024; Amini et al., 2024), which treat all preferences uniformly through binary pairwise comparisons. This binary abstraction, while effective for general language tasks, misaligns with human decision-making: humans exhibit graded preferences (*strongly* love vs. *mildly* like) and temporal sensitivity (immediate vs. delayed satisfaction), as illustrated in Figure 1. Such graded and temporally-aware patterns are pervasive in human behavior (Astington and Jenkins, 1995), yet remain largely unmodeled in current LLM-based recommenders. This raises a critical question: *what specific factors in preference data enable LLMs to capture these nuanced human behaviors for recommendation?*

We investigate this question through systematic empirical study: Our proof-of-concept experiment (§ 3) reveals that incorporating *comprehensive feedback* (including negative interactions) and *structured preference signals* (e.g., ratings) substantially

*Equal contribution
†Co-corresponding author

improves performance, suggesting binary modeling discards critical information. Through controlled ablations, we identify two key factors: **(1) preference intensity**—the graded strength of user affinity or aversion, and **(2) temporal context**—the immediacy of satisfaction. These factors, though well-established in behavioral economics and cognitive science, have been largely overlooked in LLM preference alignment for recommendation.

Building on these insights, we introduce RecPO, a preference optimization framework that operationalizes both preference intensity and temporal context through adaptive reward margins. Unlike prior work that only uses uniform margins across all preference pairs, RecPO discovers more fine-grained preference signal to model human preference: *(i)* the graded preference scores of items (e.g., 5-star vs. 3-star ratings), and *(ii)* their temporal distance from the current decision point. This enables the LLM to mirror key characteristics exhibit in human preference behaviors. Our contributions are threefold:

- We systematically demonstrate that preference intensity and temporal context are fine-grained factors for LLM-based preference modeling in recommendation, challenging the prevailing binary preference paradigm (§ 3).
- By incorporating these factors through adaptive reward margins derived from preference intensity and temporal context into LLM, we provide RecPO for sequential recommendation (§ 4).
- Through experiments on five datasets with **both explicit and implicit feedback**, we show that RecPO improves accuracy with behavioral characteristics aligned with human preference: prioritizing timely satisfaction and maintaining preference coherence under shifting contexts (§ 5).

2 Preliminaries

Existing LMs are adapted to sequential recommendation tasks through a two-stage training paradigm, namely *supervised fine-tuning (SFT)* (Ouyang et al., 2022; Liao et al., 2024; Bao et al., 2023), which adapts general-purpose LLMs into task-specific models, and *preference alignment* (Ouyang et al., 2022; Schulman et al., 2017), which further aligns model output to human preference[1].

[1]More detailed preliminaries in Appendix A.

In *SFT*, models are trained to predict the target item given users' historical interacted items along with their related contextual information. Specifically, let $\mathbf{x}_u^t$ be the task prompt that encompasses user u's interaction history up to time t, information of items to be inquired, and other task-related descriptions. Also let $\mathbf{y}_p^t$ be the text mapping of the target item that best aligns with $\mathbf{x}_u^t$'s description. The objective of SFT that optimizes π_θ is:

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

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.

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, models undergone SFT require further processing through the *preference alignment* process. One of the most prominent techniques is named DPO (Rafailov et al., 2024), which employs the Bradley-Terry (BT) model (Bradley and Terry, 1952) to model the probability of human preference data. Specifically, let $(\mathbf{y}_p, \mathbf{y}_d)$ be the relatively preferred and dispreferred textual output in a pairwise preference data, respectively. The objective is:

$$\min_\theta -\mathbb{E}_{(\mathbf{x}, \mathbf{y}_p, \mathbf{y}_d) \sim D} \Big[\log \sigma \Big(\beta \log \frac{\pi_\theta(\mathbf{y}_p|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_p|\mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_d|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_d|\mathbf{x})} \Big) \Big], \quad (2)$$

where π_{ref} is commonly set to π_{SFT}, π_θ is the aligned model, and β is a hyperparameter. Building upon DPO, a recent effort named S-DPO (Chen et al., 2024) has been proposed specifically for LLM-based recommenders. They pair each positive item with multiple negative items generated by random sampling as preference data, and revise the alignment objective as:

$$\min_\theta -\mathbb{E}_{(\mathbf{x}, \mathbf{y}_w, \mathcal{T}_d) \sim D} \Big[\log \sigma \Big(-\log \sum_{\mathbf{y}_d \in \mathcal{T}_d} \exp \Big(\beta \log \frac{\pi_\theta(\mathbf{y}_d|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_d|\mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_p|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_p|\mathbf{x})} \Big) \Big) \Big], \quad (3)$$

where $\mathcal{T}_d$ contains the item titles of multiple dispreferred items[2].

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

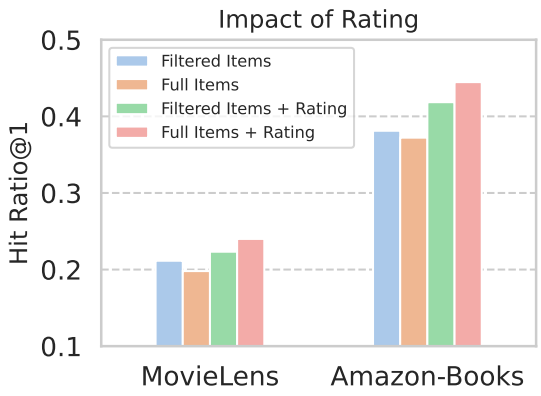


Figure 2: Hit@1 in next favorable item prediction with comprehensive and structured preference feedback.

3 What Do Current Methods Overlook? A Proof-of-Concept Investigation

Current LLM-based recommenders, including S-DPO (Chen et al., 2024), typically filter out negative feedback items from user histories and discard structured preference signals (e.g., ratings), treating all remaining items uniformly. But does this practice discard critical information?

To investigate, we design a proof-of-concept experiment that varies two dimensions of user feedback: *comprehensiveness* (whether negative interactions are retained) and *structure* (whether graded preference signals are provided). We devise four-tier input configurations that progressively integrate preference signals: (i) *Filtered Items*: Excluding negative feedback items and no explicit ratings are provided, mimicking S-DPO's setup; (ii) *Full Items*: Retaining all historical items, yet no explicit ratings are provided; (ii) *Filtered Items + Rating*: Providing explicit ratings yet excluding negative-feedback items; (iv) *Full Items + Rating*: Retaining all items and their corresponding explicit ratings.

We fine-tune LLaMA3-8B on MovieLens and Amazon-Books (described § 5.1) using the four input configurations. The experimental results are reported using Hit Ratio@1 (see § 5.1, where higher values indicate better performance) and are shown in Figure 2.

Key Findings. Three striking patterns emerge: **(1) Comprehensive feedback matters**: Comparing *Filtered Items + Rating* vs. *Full Items + Rating*, retaining negative interactions consistently improves performance. While counterintuitive—why include disliked items when predicting the next favorable item?—this suggests that aversion modeling is crucial for accurate preference profiling. **(2) Structured signals matter**: Comparing *Full Items* vs. *Full Items + Rating*, adding structured preference signals (ratings) substantially boosts performance. Without ratings, *Full Items* even underperforms *Filtered Items*, as negative items become noise without explicit annotations. **(3) Both factors are complementary**: The best performance comes from combining both—comprehensive feedback with structured signals.

These results reveal that current methods overlook two critical aspects: the *graded strength* of preferences (intensity) and the *distinction* between positive and negative feedback. However, what specific factors enable LLMs to leverage this richer information? This motivates our investigation into preference intensity and temporal context as the underlying mechanisms.

4 Methodology

We first lay out the prompt design that establishes the foundation for preference modeling in LLM-based recommendations. We then introduce RecPO, a novel preference optimization framework for sequential recommendation that dynamically calibrates reward margins between pairwise preference data based on contextualized and structured preference feedback, shown in Figure 3.

4.1 Operationalizing Comprehensive and Structured Feedback

Unlike existing approaches (Liao et al., 2024; Chen et al., 2024) which remove items with negative feedback from interaction histories to construct homogeneous sequences (i.e., items with only positive feedback), building on the observations from § 3, We preserve the complete interaction sequence for each user, explicitly retaining all historical items along with their associated preference feedback. We use explicit ratings or ratings converted from implicit feedback to construct a hierarchical preference profile. 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.

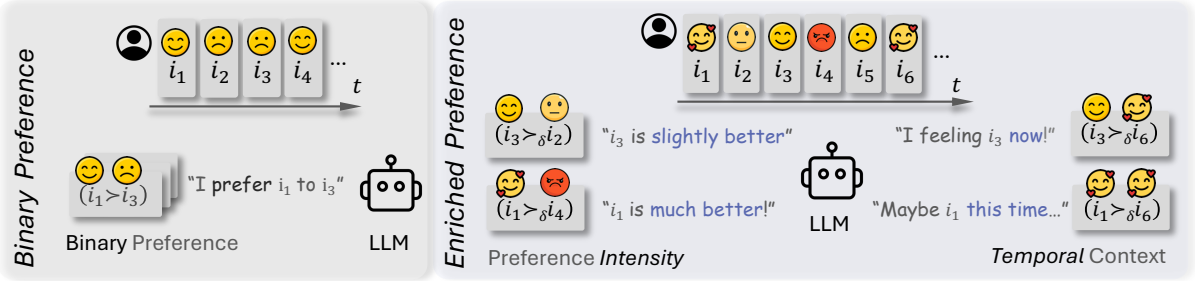


Figure 3: Illustrations for preference learning frameworks with binary and enriched preference: the prior assumes binary distinction in preference, while the latter enriches preference distinction with preference intensity and temporal context (δ indicates the enrichment).

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 We prepend the history-specific prefixes (e.g., `"Given 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. The three prompt components are concatenated as the final textual input $\mathbf{x}_u$ to the LMs. Concrete examples are demonstrated in Appendix C.

4.2 Modeling *Preference Intensity* and *Temporal Context*

Current preference learning methods simplify preference modeling to maximizing the reward difference between pairwise preferred and dispreferred responses/items, exposing them to two key limitations: *(i)* Neglecting preference intensity, where in reality, users may strongly prefer certain items while only slightly prefer others, compared to either the same or different negative items; *(ii)* Neglecting temporal context,where users typically prioritize immediate satisfaction over delayed rewards. To incorporate both factors into preference modeling, We define an adaptive target reward margin γ_r, dynamically determined by the structured preference between the two compared items and their relative recency with respect to the current timestamp. Specifically, a utility function $\phi(\cdot)$ is utilized to evaluate the reward of an item wrt two perspectives—the stronger the preference of more recent interaction, the larger the utility. The margin of a pairwise data $(\mathbf{y}_p, \mathbf{y}_d)$ is defined as:

$$\gamma_r = \lambda \frac{\phi\left(s_p, \Delta_{t_p}\right)}{\phi\left(s_d, \Delta_{t_d}\right)} \tag{4}$$

where $\mathbf{y}_p$ is preferred over $\mathbf{y}_d$, s_p and s_d are their structured preference score respectively, 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 items from either negative sampling or historical interactions where no user-assigned feedback is available, we set a default preference score and time latency to facilitate the training. More details about the default value can be found in § 5.

4.3 Deriving the Preference Alignment Objective

We plug Equation 4 into the BT model to derive the distribution for pairwise preference data:

$$P^*(\mathbf{y}_p \succ \mathbf{y}_d \mid \mathbf{x}_u) = \sigma\left(r\left(\mathbf{x}_u, \mathbf{y}_p\right) - r\left(\mathbf{x}_u, \mathbf{y}_d\right) - \gamma_r\right), \tag{5}$$

where $r(\cdot)$ is the reward function. We pair each preferred item with multiple dispreferred items, and 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 $\mathcal{C}$ 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

$$\mathcal{L}\left(\pi_\theta; \pi_{\text{ref}}\right) = -\mathbb{E}_{(\mathbf{x}_u, \mathbf{y}_p, \mathcal{T}_d)\sim\mathcal{D}}\left[\log\sigma\left(-\log\sum_{\mathbf{y}_d\in\mathcal{T}_d}\exp\left(\beta\log\frac{\pi_\theta(\mathbf{y}_d \mid \mathbf{x}_u)}{\pi_{\text{ref}}(\mathbf{y}_d \mid \mathbf{x}_u)}\right.\right.\right.$$
$$\left.\left.\left. - \beta\log\frac{\pi_\theta(\mathbf{y}_p \mid \mathbf{x}_u)}{\pi_{\text{ref}}(\mathbf{y}_p \mid \mathbf{x}_u)} - \lambda\frac{\phi(s_p, \Delta_{t_p})}{\phi(s_d, \Delta_{t_d})}\right)\right)\right]. \tag{7}$$

of observing the candidates' preference ranked as $[\mathbf{y}_{\sigma(1)}, \mathbf{y}_{\sigma(2)}, \ldots, \mathbf{y}_{\sigma(K)}]$ is:

$$P(\sigma \mid \mathbf{x}_u, \mathcal{T}_c) = \prod_{j=1}^{K}\frac{\exp\left(r\left(\mathbf{x}_u, \mathbf{y}_{\sigma(j)}\right)\right)}{\sum_{m=j}^{K}\exp\left(r\left(\mathbf{x}_u, \mathbf{y}_{\sigma(m)}\right)\right)}, \tag{6}$$

where $\mathcal{T}_c$ contains K item descriptions. Building upon Equation 6, we derive the final objective shown in Equation 7. Note that our method is reduced to S-DPO when $\lambda = 0$. For brevity, the detailed derivation process is provided in Appendix B. Optimizing the derived objective effectively integrates structured preference feedback with temporal decay factors to refine implicit preference modeling, adapting LLM recommenders to better fit preference patterns in real-world scenarios.

5 Experiment

5.1 Setup

Datasets. We select five publicly available representative recommendation benchmark datasets for our experiments: We use five widely used real-world sequential recommendation datasets for evaluation, including *MovieLens-1M* (Harper and Konstan, 2015), *Amazon-books* (Ni et al., 2019), *Steam* (Kang and McAuley, 2018), *BeerAdvocate* (Leskovec and Mcauley, 2012), and *LastFM* (Celma, 2010)[3]. For each dataset, we apply k-core filtering (He and McAuley, 2016) to remove users and items with less than $k = 5$ interactions. We 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 ratings as the structured preferences to adjust the preference margins, and for Steam and LastFM, where explicit ratings are unavailable, we rely on play-hours and play-count as proxies for user structured preferences. For each user, we order the interactions chronologically, using the second-last target interaction for validation, the last one for testing, and the rest for training.

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. We select two LM backbones, LLaMA3 (Dubey et al., 2024) and Qwen (Bai et al., 2023), and compare between the standard preference optimization baseline DPO (Rafailov et al., 2024), SimPO (Meng et al., 2024), a reference-free method that enhances DPO with length regularization and a fixed margin term, and S-DPO (Chen et al., 2024), which adapts DPO specifically for sequential recommendation[4]. Note that we **exclude** proprietary LLMs due to their lack of training access in integrating the two factors—preference intensity and temporal context—identified in our investigation. As this work is hypothesis-driven rather than solely method-focused, our objective is to validate how these factors enable effective preference modeling, which requires full control over LLMs unavailable in closed-source systems.

Implementation. All experiments are performed on 8 NVIDIA RTX A100 with 80GiB of VRAM. For all the preference learning approaches, we first conduct SFT for task adaptation, and then post-train models initialized from SFT checkpoints by optimizing the alignment loss in Equation 7[5].

Evaluation Metrics. We follow S-DPO and evaluate models using two metrics: Hit Ratio@1,

[3]More dataset details in Appendix E.1

[4]More baseline details in Appendix E.2

[5]More implementation details in Appendix E.3

Model Type	Bkbn	Method	MovieLens		Amazon-Books		BeerAdvocate		Steam		LastFM	
			HR@1	ValidRatio	HR@1	ValidRatio	HR@1	ValidRatio	HR@1	ValidRatio	HR@1	ValidRatio
Feedback Type			Explicit Feedback						Implicit Feedback			
Trad.	-	GRU4Rec	0.2664	1.0000	0.1310	1.0000	0.3708	1.0000	0.4584	1.0000	0.6630	1.0000
	-	Caser	0.2714	1.0000	0.1538	1.0000	0.3757	1.0000	0.4394	1.0000	0.6716	1.0000
	-	SASRec	0.2671	1.0000	0.1559	1.0000	0.3800	1.0000	0.4587	1.0000	0.6659	1.0000
LLM	LLaMA3-8B	LLaMA3	0.0929	0.7351	0.0654	0.6165	0.0686	0.6617	0.0852	0.8672	0.1264	0.6147
		SFT	0.2478	0.9985	0.4447	0.9974	0.2645	0.9936	0.3122	0.9990	0.5076	1.0000
		DPO	0.2809	0.9970	0.5049	0.9887	0.4412	0.9875	0.3340	0.9980	0.5719	1.0000
		SimPO	0.2974	0.9725	0.5129	0.9564	0.4020	0.9250	0.3401	0.9766	0.5759	0.9419
		S-DPO	0.2902	0.9983	0.5065	0.9880	0.4698	0.9903	0.3588	0.9990	0.5719	0.9990
		RecPO	**0.3451**	0.9969	**0.5802**	0.9851	**0.5771**	0.9887	**0.4672**	0.9985	**0.6830**	0.9959
	Qwen-7B	Qwen	0.1204	0.7471	0.1013	0.7194	0.0583	0.4223	0.1477	0.6293	0.2148	0.6860
		SFT	0.2060	0.9983	0.3659	0.9967	0.2044	0.9849	0.2081	0.9950	0.3119	0.9969
		DPO	0.2610	0.9983	0.4412	0.9930	0.2600	0.9724	0.2457	0.9960	0.4046	0.9969
		SimPO	0.2888	0.9531	0.4644	0.9880	0.4044	0.9529	0.3706	0.9940	0.5209	0.9796
		S-DPO	0.2706	0.9957	0.4623	0.9910	0.3253	0.9798	0.3062	0.9970	0.4495	0.9959
		RecPO	**0.3446**	0.9896	**0.5307**	0.9880	**0.4320**	0.9729	**0.4143**	0.9912	**0.5973**	0.9980

Table 1: Overall model performance comparison on five real-world recommendation datasets. The best performance is bolded, and runner-ups are underlined. Datasets are grouped by explicit and implicit feedback.

which measures the proportion of test cases where the top-ranked item matches the ground-truth target, and Valid Ratio, which captures instruction compliance by quantifying the fraction of outputs that follow formatting rules and remain within the candidate set. The latter ensures outputs are valid and in-distribution. Together, they assess both recommendation accuracy and practical deployability.

5.2 Do Preference Intensity and Temporal Context Improve Recommendation?

Overall Performance. Table 1 compares RecPO with the baselines across the five datasets, revealing the following key findings:

- ***SFT establishes base model capabilities.*** While LLMs 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, matching traditional recommenders and demonstrating the necessity to align general-purpose LLMs with specific behavioral requirements in real-world applications.

- ***Binary preference modeling provides limited gains.*** All preference learning methods, including our proposed RecPO, DPO, SimPO, and S-DPO, significantly outperform SFT in Hit Ratio@1, suggesting the alignment between complex preferences 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 an 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.

- ***Integrating preference intensity and temporal context yields substantial improvements.*** By integrating structured preference with contextualization adaptive reward margins, RecPO universally improves Hit Ratio@1 over other LLM-based approaches across both of the language backbones. This can be attributed to its human-aligned preference modeling grounded in cognitive science principles that generalize well across instructive learning tasks. Compared to traditional recommenders on implicit feedback datasets, the performance gains of RecPO are relatively modest. We posit that this narrower gap arises from proxy-derived preference signals, which exhibit homogeneous interaction patterns that even simple traditional models can effectively capture.

How Should Preference Intensity and Temporal Context Be Combined? 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 4. To investigate how these factors should be mathematically combined, we introduce two alternative margin functions: *(i)*

Dataset	Log Diff	Log Ratio	RecPO
MovieLens	0.3160	0.3247	0.3451
Amazon-Books	0.5370	0.5455	0.5802
BeerAdvocate	0.5023	0.5257	0.5771
Steam	0.4284	0.4517	0.4672
LastFM	0.5912	0.6388	0.6830

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

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 2, both variants outperform the strongest LLM-based recommender baseline, confirming that incorporating these factors through any margin formulation improves performance. RecPO 's default ratio-based margin achieves the best overall performance by amplifying training gradients, especially when historical user ratings show low volatility. By directly contrasting ϕ_p and ϕ_d via division, it provides stronger learning signals that help the model prioritize subtle but critical preference patterns.

5.3 Do Models Learn Human-Aligned Preference Patterns?

Beyond quantitative performance, we investigate whether incorporating preference intensity and temporal context enables models to exhibit human-like decision patterns. We probe the learned preferences from multiple perspectives:

- **Temporal context sensitivity**: When the candidate set includes other future highly-rated items, does the model still prioritize the correct next item, reflecting sensitivity to temporal context?
- **Preference intensity awareness**: When the candidate set includes future low-rated items that may appear contextually tempting, can the model avoid recommending them?
- **Implicit aversion modeling**: When directly prompted, can the model correctly identify the item least aligned with the user's preferences?
- **Robustness across contexts**: Does the model maintain stable performance across users with varying lengths of interaction history?

Temporal context enables immediate satisfaction prioritization. To assess RecPO 's ability to model contextualized preferences, we construct more challenging test sets for MovieLens and Amazon-Books by augmenting the candidate pool with other highly-rated items from users' future interactions. This setup tests whether the model can prioritize the correct next item when competing items, though eventually preferred, are not immediately relevant. We quantify this behavior using the *Adherence Rate* (detailed in Appendix E.4), which measures h*ow often the model recommends the next immediately preferred item over delayed but highly rated alternatives*. As shown in Figure 4(a), RecPO consistently outperforms both SFT and S-DPO, more reliably ranking the temporally appropriate item at the top. This suggests improved sensitivity to short-term intent and temporal alignment. In contrast, S-DPO fails to consistently outperform SFT, indicating a failure to fully capture context-dependent user goals. Overall, RecPO 's adaptive reward margins leads to recommendations that more faithfully reflect temporally grounded human preferences.

Preference intensity enables discernment under temptation. Beyond modeling contextualized user preferences, we evaluate the model's ability to avoid recommending items that are ultimately dispreferred, even when they appear contextually relevant. To this end, we construct test sets from MovieLens and Steam by augmenting candidate sets with low-rated items from users' future interactions. While these items are rated poorly in hindsight, their later occurrence, often driven by exposure or curiosity, makes them superficially plausible as next-item recommendations, thus posing a form of contextual temptation. We use the *Avoidance Rate* (detailed in Appendix E.4) to measure the model's ability to *reject such items* when predicting the next interaction. As shown in Figure 4(b), RecPO consistently achieves the highest avoidance rates across benchmarks, outperforming all baselines. These results indicate that incorporating structured feedback enables the model to internalize both positive and negative preference signals—reducing the likelihood of recommending irrelevant or disliked items, and thereby enhancing overall alignment with user intent.

Both factors jointly enable implicit aversion modeling. While most preference alignment focuses on promoting desirable items, an essential aspect of human-like decision-making is the ability to deliberately avoid dispreferred options. To evaluate this capacity, we construct a test set querying the model directly at inference time to identify the item least aligned with a user's preferences, without providing any explicit supervision for aversion.

(a) Contextualized Preference Adherence (b) Unfavorable Item Avoidance (c) User Aversion Modeling (d) Sequence Length Robustness

Figure 4: Comparing between SFT, S-DPO, and RecPO from the perspectives of adhering to contextual preference (a), avoiding unfavorable items under temptation (b), identifying dis-preferred items (c), and consistently performing across varying user history lengths (d). The adherence rate and avoidance rate are defined in § 5.3.

This setup tests whether the model's learned preference representation implicitly encodes negative signals alongside positive ones. As shown in Figure 4(c), RecPO consistently outperforms SFT and S-DPO with higher aversion accuracy across both datasets. This suggests that RecPO internalizes a more complete structure of user preferences, capable of both attraction and avoidance. Notably, **this behavior emerges without explicit aversion labels**: through alignment with structured and contextualized feedback alone, RecPO learns to infer which items users are likely to reject.

Learned patterns generalize across varying interaction contexts. In Figure 4(d), we investigate RecPO 's robustness to variations in historical interaction lengths using the BeerAdvocate dataset. We partition the test set into subsets based on the number of past interactions and evaluate performance within each group. RecPO exhibits sustained efficacy, consistently outperforming SFT and S-DPO with larger margins. While all models follow similar performance trends as history length increases, RecPO exhibits the greatest stability, with the lowest variance in Hit Ratio@1 (8.7% vs. 17.8% for S-DPO). These results highlight RecPO 's adaptability to diverse context—a critical trait for real-world systems where user histories vary widely.

6 Related Work

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 (Liao et al., 2024; Bao et al., 2023; Geng et al., 2022; Yuan et al., 2023).

LLM preference alignment techniques aim to align language models' outputs with human preferences. Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) and DPO (Rafailov et al., 2024) fine-tune models using pairwise 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. Most recently, S-DPO (Chen et al., 2024) adapts alignment for sequential recommendation using list-wise negative items. However, these methods treat all preferences uniformly through binary comparisons, overlooking graded preference intensity and temporal context. More details are in Appendix D.

Our work differs by investigating what factors enable effective preference modeling in LLM-based recommendation, revealing that preference intensity and temporal context are critical yet overlooked dimensions.

7 Conclusion

We investigate what makes LLMs effective at modeling user preferences in sequential recommendation. Through systematic empirical study, we identify two critical factors overlooked by current binary preference modeling: *preference intensity*—the graded strength of user affinity or aversion—and *temporal context*—the immediacy of satisfaction. In contrast, prior methods that rely on binary modeling discard essential information. Motivated by these findings, we propose RecPO, which operationalizes these factors through adaptive reward margins. Our SoTA results demonstrate that preference intensity and temporal context are fundamental to effective LLM-based recommendation, with implications for human preference modeling.

Limitations

While our results demonstrate that incorporating comprehensive and structured interaction feedback improves user preference profiling, this work adopts a simplified, sequential preference structure and considers only satisfaction delay as the contextual factor. In reality, human decision-making reflects more complex hierarchies and richer contextual influences. Future research should explore how to model cognitively plausible preferences across broader preference-based tasks, extending beyond recommendations. Even within the recommendation domain, evaluations should move beyond single metrics, aiming to capture more holistic and behaviorally grounded patterns of user preference.

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*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, et al. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

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*.

Oscar Celma. 2010. Music recommendation and discovery in the long tail. Technical report, Universitat Pompeu Fabra.

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*.

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*.

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*.

F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*.

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*.

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

Jure Leskovec and Julian Mcauley. 2012. Learning to discover social circles in ego networks. In *NeurIPS*.

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*.

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

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP-IJCNLP*.

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*.

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*.

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 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 the subset of interactions up to time t, and let i_p^{t+} denote the ***next recent favorable (highly-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 $\mathbf{y}_p^t$ 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:

$$\max_\theta \quad \mathbb{E}_{(\mathbf{x}_u^t, \mathbf{y}_p^{t+}) \sim \mathcal{D}_{\text{SFT}}} \left[\log \pi_\theta(\mathbf{y}_p^{t^+} | \mathbf{x}_u^t) \right]. \quad (7)$$

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 their 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:

$$\max_\theta \quad \mathbb{E}_{x \sim \mathcal{D}, \mathbf{y} \sim \pi_\theta(\cdot|\mathbf{x})} \Big([r(\mathbf{x}, \mathbf{y})] - \beta D_{\text{KL}} \left[\pi_\theta(\mathbf{y}|\mathbf{x}) \| \pi_{\text{ref}}(\mathbf{y}|\mathbf{x}) \right] \Big), \quad (8)$$

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 8, 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:

$$\min_\theta - \mathbb{E}_{(\mathbf{x}, \mathbf{y}_w, \mathbf{y}_l) \sim D} \Big[\log \sigma \Big(\beta \log \frac{\pi_\theta(\mathbf{y}_w|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_w|\mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_l|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_l|\mathbf{x})} \Big) \Big]. \quad (9)$$

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) has been 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:

$$\min_\theta - \mathbb{E}_{(\mathbf{x}, \mathbf{y}_w, \mathcal{T}_d) \sim D} \Big[\log \sigma \Big(- \log \sum_{\mathbf{y}_d \in \mathcal{T}_d} \exp \Big(\beta \log \frac{\pi_\theta(\mathbf{y}_d|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_d|\mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_p|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_p|\mathbf{x})} \Big) \Big) \Big], \quad (10)$$

where $\mathcal{T}_d$ contains the item titles of multiple dispreferred items[6].

B Derivation of Preference Distribution

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

$$P(\mathbf{y}_i \succ \mathbf{y}_j) = \sigma\left(r\left(\mathbf{x}, \mathbf{y}_w\right) - r\left(\mathbf{x}, \mathbf{y}_l\right)\right) = \frac{\exp\left(r\left(\mathbf{x}_u, \mathbf{y}_i\right)\right)}{\exp\left(r\left(\mathbf{x}_u, \mathbf{y}_i\right)\right) + \exp\left(r\left(\mathbf{x}_u, \mathbf{y}_i\right)\right)}, \tag{11}$$

where $r(\cdot)$ is the reward model. 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 the pairwise probability is defined as

$$P(\mathbf{y}_i \succ \mathbf{y}_j) = \frac{w_i \exp\left(-\gamma_{ij}\right)}{w_i \exp\left(-\gamma_{ij}\right) + w_j} \tag{12}$$

where we assume $\gamma_{ij} = -\gamma_{ji}$. Specifically, we can use the Plackett-Luce model to decompose 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:

$$P(i \text{ chosen from } \mathcal{C}) = \frac{w_i \exp\left(-\sum_{j \in \mathcal{C} \setminus \{i\}} \gamma_{ij}\right)}{\sum_{k \in \mathcal{C}} w_k \exp\left(-\sum_{j \in \mathcal{C} \setminus \{k\}} \gamma_{kj}\right)}. \tag{13}$$

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

$$\mathcal{C}_r = \mathcal{C} \setminus \{\sigma(1), \sigma(2), \ldots, \sigma(r-1)\} \tag{14}$$

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

$$P(\sigma(r) \mid \sigma(1), \ldots, \sigma(r-1)) = \frac{w_{\sigma_{(r)}} \exp\left(-\sum_{j \in \mathcal{C}_r \setminus \{\sigma(r)\}} \gamma_{\sigma(r)j}\right)}{\sum_{k \in \mathcal{C}_r} w_{\sigma_{(k)}} \exp\left(-\sum_{j \in \mathcal{C}_r \setminus \{k\}} \gamma_{kj}\right)}. \tag{15}$$

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

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

$$P(\sigma \mid \mathcal{C}) = \prod_{r=1}^{K-1} \frac{w_{\sigma_{(r)}} \exp\left(-\sum_{j \in \mathcal{C}_r \setminus \{\sigma(r)\}} \gamma_{\sigma(r)j}\right)}{\sum_{k \in \mathcal{C}_r} w_{\sigma_{(k)}} \exp\left(-\sum_{j \in \mathcal{C}_r \setminus \{k\}} \gamma_{kj}\right)} \tag{16}$$

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 the recommendation case given interactions $\mathbf{x}_u$ of user u, multiple negative items $\mathbf{y}_d \in \mathcal{T}_d$, and the positive item $\mathbf{y}_p$:

$$P(\mathbf{y}_p \succ \mathbf{y}_d, \forall \mathbf{y}_d \in \mathcal{T}_d \mid \mathbf{x}_u, \mathbf{y}_p, \mathcal{T}_d) = \frac{w_p \exp\left(-\sum_{j=1}^{K-1} \gamma_{p,d_j}\right)}{w_p \exp\left(-\sum_{j=1}^{K-1} \gamma_{p,d_j}\right) + \sum_{j=1}^{K-1} w_{d_j}}. \tag{17}$$

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

C Prompt Examples

We refer to the prompts used in previous works (Chen et al., 2024; Liao et al., 2024) to construct prompts utilized in our work. Examples in Figure 5 demonstrate the prompts for sequential recommendation.

D 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. Convolution-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.

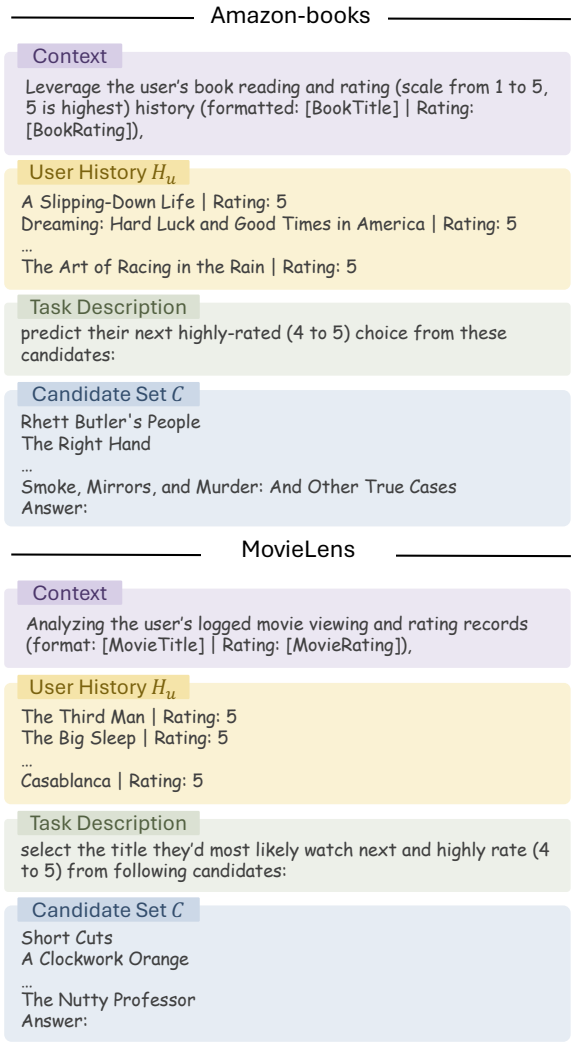


Figure 5: Textual prompt examples for Amazon-books and MovieLens.

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
LastFM	982	107,296	307,829

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.

E Experimental Settings

E.1 Datasets

We use five widely used real-world sequential recommendation datasets for evaluation, including *MovieLens-1M*[7] (Harper and Konstan, 2015), *Amazon-books*[8] (Ni et al., 2019), *Steam*[9] (Kang

[7] https://grouplens.org/datasets/movielens/1m/
[8] https://nijianmo.github.io/amazon/index.html
[9] https://github.com/kang205/SASRec

and McAuley, 2018), *BeerAdvocate*[10] (Leskovec and Mcauley, 2012), and *LastFM*[11] (Celma, 2010). We demonstrate the dataset statistics in Table 3. The MovieLens-1M dataset is sourced from the MovieLens platform and contains 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. The BeerAdvocate dataset collects beer reviews that cover multiple sensory aspects along with overall ratings. The LastFM dataset comprises detailed music listening records for nearly 1,000 users, including user profiles with demographic information, artist and track identifiers, and precise timestamps for each listening event.

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 preference profiling.

For Steam and LastFM, since they lack explicit rating signals, we convert play-hours and play-count, respectively, to a 1-to-5 scale structured rating based on their 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.

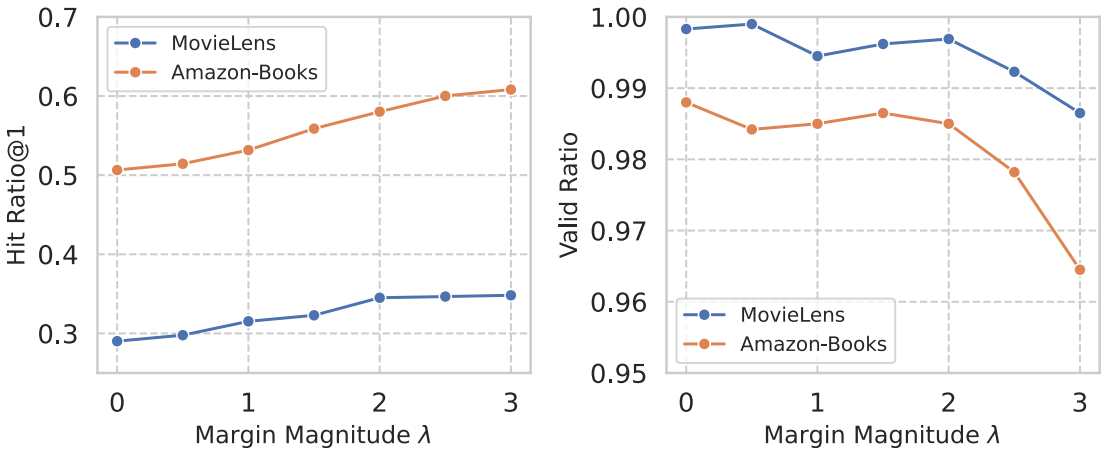
[10] https://cseweb.ucsd.edu/~jmcauley/datasets.html#multi_aspect

[11] http://ocelma.net/MusicRecommendationDataset/lastfm-1K.html

E.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 for 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.
- Qwen2.5 (Bai et al., 2023) is a recent LLM developed by Alibaba, optimized for instruction-following and multi-turn dialogue tasks.
- 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.



(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.

E.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) and Qwen2.5-7B (Bai et al., 2023) as the base models 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, 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 C, 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.

E.4 Evaluation Metrics

As mentioned in Section 5.1, 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 5.3, 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. Note that we use rating to denote the preference hierarchy, yet it can be derived from either implicit or explicit feedback. **In the contextualized preference adherence experiment, the candidate set for testing includes at least two highly-rated items from the subsequent sequence**. We follow the rule described in Section 2 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 highly-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.

F Analysis on Margin Magnitude

As detailed in Section 4, the parameter λ controls the 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.