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

Zhongyu Ouyang[1*] **Qianlong Wen**[2*‡] **Chunhui Zhang**[1]
Yanfang Ye[3] **Soroush Vosoughi**[1†]
[1]Department of Computer Science, Dartmouth College
[2] ByteDance, US
[3]Department of Computer Science and Engineering, University of Notre Dame
{zhongyu.ouyang.gr, chunhui.zhang.gr, soroush.vosoughi}@dartmouth.edu
qianlong.wen@bytedance.com, yye7@nd.edu

Abstract

What enables large language models (LLMs) to effectively model user preferences in sequential recommendation? Our investigation reveals that existing preference-alignment approaches largely rely on binary pairwise comparisons, overlooking two critical factors: *preference intensity*—the structured strength of affinity or aversion —and *temporal context*—the extent to which recent interactions better reflect a user's current intent. Through controlled experiments, we show that leveraging comprehensive feedback with structured preference signals substantially improves recommendation performance, indicating that binary modeling discards essential information. Motivated by these findings, we propose RecPO, a unified preference optimization framework that maps both explicit and implicit feedback into a common preference signal and constructs adaptive reward margins that jointly account for preference intensity and interaction recency. Experiments across five datasets show that RecPO consistently outperforms state-of-the-art baselines while exhibiting behavioral patterns aligned with human decision-making, including favoring immediate satisfaction, maintaining preference coherence, and avoiding dispreferred items. Our results highlight that preference intensity and temporal context are fundamental ingredients for effective LLM-based recommendation. Code: https://github.com/zyouyang/RecPO

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 (Chen et al., 2024a) leverage semantic understanding and reasoning capabilities to model user preferences from textual interaction histories from multiple perspectives.

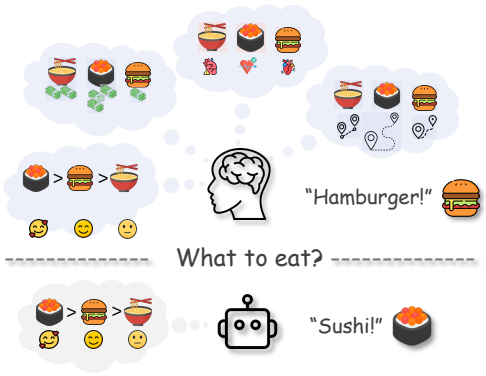

Figure 1: Human decision-making involves trade-offs among *preference intensity, temporal context, effort, and risk*—factors that are largely overlooked in current LLM-based preference modeling.

Current approaches predominantly rely on preference alignment techniques such as DPO (Rafailov et al., 2024) and its variants (Chen et al., 2024b; 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 structured preferences (*strongly* love vs. *mildly* like) and recency-sensitive temporal preferences, where more recent interactions better reflect current intent than older ones, as illustrated in Figure 1. Such structured and temporally contextualized patterns are pervasive in human behavior (Astington and Jenkins, 1995), yet remain 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

*Equal contribution. †Corresponding author.
‡ Work done during PhD at University of Notre Dame.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14417–14434
July 2-7, 2026

empirical study. Our proof-of-concept experiment (§ 3) reveals that incorporating *comprehensive feedback* (including negative interactions) and *structured preference signals* (e.g., ratings) substantially improves performance, indicating that binary modeling discards critical information. Through controlled ablations, we identify two key factors: **(1) preference intensity**—the structured strength of user affinity or aversion, and **(2) temporal context**—the extent to which more recent interactions better reflect a user's current intent. 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 preference intensity and temporal context via adaptive reward margins. Unlike prior work that applies uniform margins across all preference pairs, RecPO leveraged fine-grained preference signals—*(i)* structured preference strength (e.g., 5-star vs. 3-star ratings), and *(ii)* interaction recency relative to the current decision point—to modulate preference alignment. As a result, RecPO enables LLMs to model evolving user preferences in a manner more consistent with human decision-making. Our contributions are threefold:

- We systematically demonstrate that preference intensity and recency-sensitive temporal context are critical factors for LLM-based preference modeling in sequential recommendation(§ 3).
- We propose RecPO, a unified preference optimization framework that incorporates these factors via adaptive reward margins, enabling effective preference alignment for sequential recommendation (§ 4).
- Through experiments on five datasets with both explicit and implicit feedback, we show that RecPO improves recommendation accuracy and exhibits human-aligned behaviors by prioritizing items aligned with current user intent and maintaining coherent preferences under shifting contexts (§ 5).

2 Preliminaries

Sequential Recommendation with LMs. We begin by formalizing the sequential recommendation task within the LM framework. 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 (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+}.

Adapting and Aligning LMs with Human Preference Feedback. 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* (Schulman et al., 2017; Ouyang et al., 2022), which further aligns model output to human preference[1].

In *SFT*, the model is trained to predict the target item given a user's interaction history and contextual information. Let $\mathbf{x}_u^t$ be the task prompt constructed from user u's interactions up to time t, and let $\mathbf{y}_p^t$ be the textual mapping of the target item. The SFT objective 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 resulting model is denoted as π_{SFT}. For brevity, we omit time superscripts when unambiguous.

While SFT adapts LMs to the task format, 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, preference alignment further optimizes models using preference data. A representative method is DPO (Rafailov et al., 2024), which models pairwise preferences using the Bradley–Terry framework (Bradley and Terry, 1952). Given a preferred–dispreferred output pair

[1]More detailed preliminaries in Appendix A.

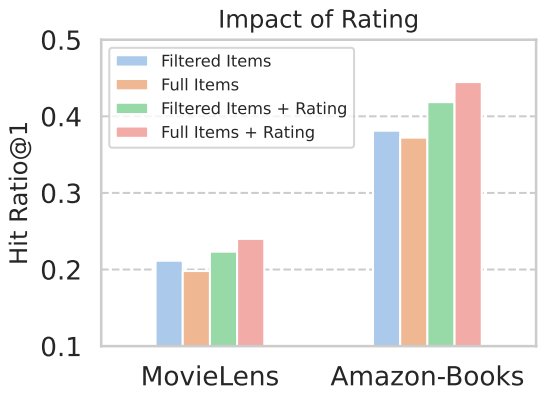

Figure 2: Hit@1 in next favorable item prediction with comprehensive and structured preference feedback.

$(\mathbf{y}_p, \mathbf{y}_d)$, the DPO 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 typically set to π_{SFT} and β controls the strength of preference alignment.

For sequential recommendation, S-DPO (Chen et al., 2024b) extends DPO by pairing each preferred item with multiple dispreferred items $\mathcal{T}_d$, yielding the objective:

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

where $\mathcal{T}_d$ denotes the set of dispreferred items[2].

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

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

To investigate this, we design a proof-of-concept experiment that varies two dimensions of user feedback: *comprehensiveness* (whether negative interactions are retained) and *structure* (whether structured preference signals are provided). We consider four input configurations: (i) *Filtered Items*: excluding negative interactions, without explicit ratings (mimicking S-DPO's setup); (ii) *Full Items*: retaining all interactions, without ratings; (iii) *Filtered Items + Rating*: excluding negative interactions, with ratings; (iv) *Full Items + Rating*: retaining all interactions with corresponding ratings.

Using SFT only, we fine-tune LLaMA3-8B on MovieLens and Amazon-Books (described § 5.1) under each configuration. Performance is evaluated using Hit Ratio@1 (see § 5.1, higher is better), with results shown in Figure 2.

Key Findings. **(1) Comprehensive feedback enables aversion modeling.** Comparing *Filtered Items + Rating* vs. *Full Items + Rating*, retaining negative interactions consistently improves performance. This indicates that negative interactions provide informative aversion signals that help delineate user preferences when their strength is explicitly encoded. **(2) Structured signals are required to recover preference intensity.** Comparing *Full Items* vs. *Full Items + Rating*, adding structured preference annotations yields substantial gains. Without such signals, *Full Items* underperforms *Filtered Items*, as unannotated negative interactions collapse into noise. **(3) Both factors are jointly necessary.** The best performance is achieved only when comprehensive feedback is combined with structured preference signals, suggesting that effective preference modeling requires access to both the full interaction spectrum and structured preference information.

These results indicate that existing methods overlook the aspects of structured strength of preferences (preference intensity) and the informative role of negative interactions in delineating user aversion. A natural question then arises: how can LLMs effectively exploit such fine-grained signals when they are embedded within a user's interaction sequence? Crucially, when preferences are modeled as structured signals over time, their contribution to future decisions becomes nonstationary. Interactions with similar preference intensity may differ substantially in relevance depending on their temporal proximity to the current decision. This observation motivates our study of **preference intensity** and **recency-sensitive temporal context** as complementary factors for enabling LLMs to capture richer, human-aligned preference dynamics.

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

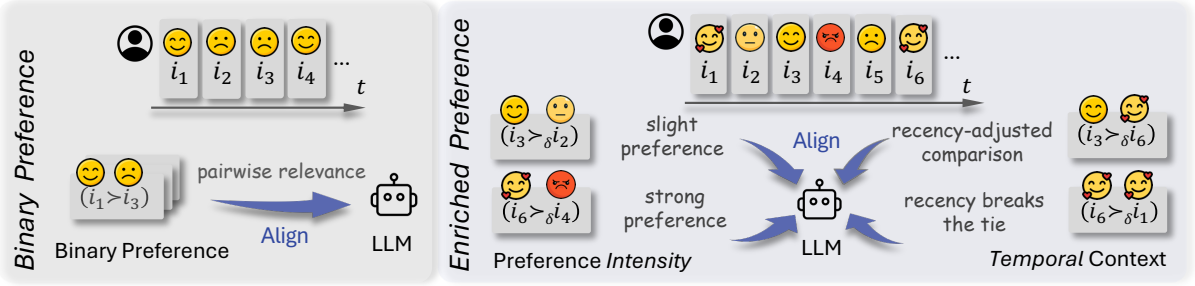

Figure 3: Illustrations for preference learning frameworks with binary and enriched preference: the prior assumes binary distinction in preference, while the latter enriches preference distinction with preference intensity and temporal context (δ indicates the enrichment).

4 Methodology

We first lay out the prompt design underlying preference modeling in LLM-based recommendation, and then present RecPO, a preference optimization framework that calibrates reward margins using structured and temporal contextualized preference feedback (Figure 3).

4.1 Operationalizing Comprehensive and Structured Feedback

Instead of discarding negatively interacted items to construct homogeneous histories (Liao et al., 2024; Chen et al., 2024b), we retain each user's complete interaction sequence, including both positive and negative feedback. Each historical item is paired with an associated preference signal, obtained either from explicit ratings or from implicit feedback converted into structured scores, yielding a structured preference profile. Following prior work (Chen et al., 2024b), the input prompt is composed of the following components:

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 preference 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 will 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*

We extend standard pairwise preference optimization by introducing an adaptive reward margin γ_r that is determined by both the structured preference strength of the compared items and their relative recency with respect to the current decision point. Specifically, we define γ_r for a preference pair $(\mathbf{y}_p, \mathbf{y}_d)$ 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 $\mathbf{y}_d$, $\Delta_{t_p} = t_p^+ - t$ represents the temporal distance between the interaction and the current decision point, and λ controls the margin magnitude. The terms s_p and s_d are their structured preference scores derived from explicit or implicit feedback; they are defined abstractly and may incorporate multiple forms of preference evidence rather than being tied to a single signal. The utility function $\phi(\cdot)$ *jointly* captures preference intensity and temporal relevance.

In this work, we instantiate $\phi\left(s, \Delta_t\right) = s/\left(\Delta_t\right)^{0.5}$, though alternative forms of the general family $\phi(s, \Delta_t) \propto s/(\Delta_t)^{\alpha}$ with $\alpha > 0$ are also compatible. For dispreferred items from negative sampling or historical interactions without explicit user feedback, we assign default preference scores and time latencies to facilitate training. More implementation details are provided in § 5.

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

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, and $\sigma(\cdot)$ is the sigmoid 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 τ representing the predicted ranking of these candidates based on user preference for the next item (denote $\tau(j)$ as the item ranked at position j), the probability of observing the candidates' preference ranked as $[\mathbf{y}_{\tau(1)}, \mathbf{y}_{\tau(2)}, \ldots, \mathbf{y}_{\tau(K)}]$ is:

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

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 intensity with recency-sensitive temporal context, enabling LLM recommenders to align preference learning with users' evolving intent in sequential decision settings.

5 Experiment

5.1 Setup

Datasets. 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). More dataset details are provided in Appendix E.1.

For each dataset, we apply k-core filtering (He and McAuley, 2016) with $k = 5$ to remove users and items with insufficient interactions. We construct a candidate set of 20 items for next-item prediction. During training, the candidate set includes 10 subsequent interactions (ensuring the ground-truth item is present) and 10 randomly sampled non-interacted items; during validation and testing, the set consists of the ground-truth item and 19 randomly sampled non-interacted items. For MovieLens-1M, Amazon-Books, and BeerAdvocate, explicit ratings are used as structured preference signals, while for Steam and LastFM, where explicit ratings are unavailable, we use play-hours and play-count as proxies. For each user, interactions are ordered chronologically, with the second-last interaction used for validation, the last for testing, and the remainder 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., 2024b), which adapts DPO specifically for sequential recommen-

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.

dation. More baseline baseline details are provided in Appendix E.2.

We *exclude* proprietary LLMs due to their lack of training access in integrating preference intensity and temporal context. As this work is hypothesis-driven rather than 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[3].

Evaluation Metrics. We follow S-DPO and evaluate models using two metrics: (i) *Hit Ratio@1* measures the proportion of test cases where the top-ranked item matches the ground-truth target, and (ii) *Valid Ratio* 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.*** Raw LLMs, while possess rich world knowledge, frequently violate recommendation constraints. SFT substantially improves output validity, matching traditional recommenders and underscoring the need for task-specific adaptation.

- ***Binary preference modeling provides limited gains.*** Preference optimization methods consistently outperform SFT in Hit Ratio@1, but standard DPO provides modest improvements. Methods that incorporate multiple negatives (S-DPO and RecPO) perform better, while SimPO achieves higher accuracy at the cost of reduced output validity. This suggests that treating preferences as isolated binary comparisons underutilizes available preference information in sequential recommendation.

- ***Integrating preference intensity and temporal context yields substantial improvements.*** By integrating structured preference signals with recency-sensitive adaptive margins, RecPO consistently achieves the best performance across both LLM backbones. Gains over traditional recommenders are smaller on implicit-feedback datasets, likely due to the homogeneity of proxy-derived preference signals that even simple traditional models can effectively capture.

Do Preference Intensity and Temporal Context Both Matter? We ablate the contributions of preference intensity and recency-sensitive temporal context by progressively removing components

[3]More implementation details in Appendix E.3

Hit Ratio@1					
Variant	ML	AmazonB	Beer	Steam	LastFM
–I –T	0.2902	0.5065	0.4698	0.3588	0.5719
–T	0.3343	0.5661	**0.6143***	0.4202	0.6544
RecPO	**0.3451**	**0.5802**	0.5771	**0.4672**	**0.6830**
Valid Ratio					
Variant	ML	AmazonB	Beer	Steam	LastFM
–I –T	0.9983	0.9880	0.9903	0.9990	0.9990
–T	0.9983	0.9798	0.9407*	0.9980	0.9959
RecPO	0.9969	0.9851	0.9887	0.9985	0.9959

Table 2: Ablation study on preference intensity and recency-sensitive temporal context. –I –T removes adaptive margins (equivalent to S-DPO); –T uses preference intensity only; **Excessively low valid ratios are impractical for real-world deployment.*

of the adaptive margin. As shown in Table 2, removing the margin entirely (–I –T), which reduces RecPO to S-DPO, leads to consistent performance drops across datasets. Incorporating preference intensity alone (–T) substantially improves accuracy, confirming the importance of structured preference signals. RecPO achieves the best overall performance on most datasets, demonstrating that preference intensity and temporal context are complementary.

How Should Preference Intensity and Temporal Context Be Combined? Let ϕ_p and ϕ_d be the scores for the preferred and dispreferred items respectively. By default, RecPO defines the margin term γ_t as the ratio of preference scores ϕ between positive and negative item pairs (Equation 4). To investigate how these factors should be mathematically combined, 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 3, both variants outperform the strongest LLM-based recommender baseline, confirming the benefits of incorporating these factors through any margin formulation. 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, stronger learning signals are provided to 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:

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

- **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. To this end, we define the *Adherence Rate* as the proportion of cases in which the model recommends the next immediately preferred item. A high Adherence Rate indicates that the model consistently recommends relevant and contextualized items among all highly-rated candidates. A higher Avoidance Rate signifies that the model is better at resisting the temptations from the unfavorable items. More details are included in Appendix E.4.

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

(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.

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. To measure how well a model resists such temptations when predicting the next interaction, we define the *Avoidance Rate* as the proportion of cases in which the model successfully avoids recommending them. More metric details are provided in Appendix E.4.

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 while consistently adhering to user preference trajectories, 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. This setup tests whether the model's learned preference representation implicitly encodes negative signals alongside positive ones. We define aversion accuracy as the proportion of cases in which the model correctly identifies the least preferred item without explicit aversion supervision.

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 items users are most 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.

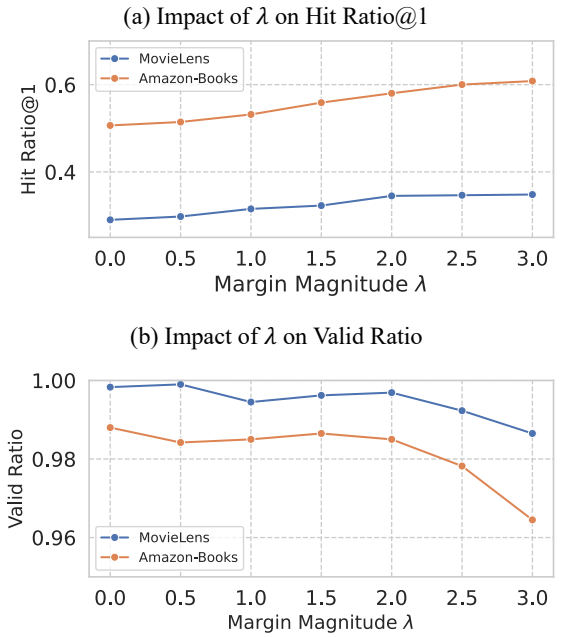

Figure 5: Sensitivity analysis of the margin parameter λ on: (a) Hit Ratio@1 and (b) Valid Ratio across MovieLens and Amazon-Books datasets.

6 Analysis on Margin Magnitude

As detailed in § 4, 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 5. 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.

7 Related Work

Sequential recommendation models temporal user preferences in interaction sequences, different from general recommendation tasks that treat user behavior as static and order-independent signals (Rendle et al., 2009; He et al., 2020; Wu et al., 2021; Ouyang et al., 2025a,b, 2024). Early methods adopt structures such as recurrent neural networks (Hidasi, 2016), self-attention mechanisms (Kang and McAuley, 2018), and convolutional layers (Chang et al., 2021) for temporal modeling. Further methods further advance 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. Recent advances integrate LLMs for their rich semantic understanding and contextual reasoning capabilities (Liao et al., 2024; Bao et al., 2023; 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., 2024b) adapts alignment for sequential recommendation using list-wise negative items. However, these methods model preferences through binary comparisons, overlooking structured 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.

8 Conclusion

We investigate what enables LLMs to effectively model user preferences in sequential recommendation. Our analysis identifies two key factors—reference intensity and recency-sensitive temporal context—that are largely collapsed by binary preference modeling. Motivated by these findings, we propose RecPO, a preference optimization framework that operationalizes these factors via adaptive reward margins. Extensive experiments demonstrate that jointly modeling preference intensity and temporal context yields consistent improvements over state-of-the-art baselines. More broadly, this work advocates a shift in LLM-based recommendation away from static, uniform preference supervision toward learning from richer interaction traces and dynamic contextual signals as first-class sources of preference information.

Acknowledgments

This research was supported in part by the National Science Foundation under Grant No. 2452367.

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*.

Jizheng Chen, Kounianhua Du, Jianghao Lin, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024a. Elcorec: Enhance language understanding with co-propagation of numerical and categorical features for recommendation. In *CIKM*.

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. 2024b. On softmax direct preference optimization for recommendation. In *NeurIPS*.

Xingjian Diao, Zheyuan Liu, Chunhui Zhang, Weiyi Wu, Keyi Kong, Lin Shi, Kaize Ding, Soroush Vosoughi, and Jiang Gui. 2026. Addressing overthinking in large vision-language models via gated perception-reasoning optimization. *arXiv preprint arXiv:2601.04442*.

Xingjian Diao, Chunhui Zhang, Keyi Kong, Weiyi Wu, Chiyu Ma, Zhongyu Ouyang, Peijun Qing, Soroush Vosoughi, and Jiang Gui. 2025. Soundmind: Rl-incentivized logic reasoning for audio-language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

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*.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *ACM SIGIR conference*.

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

Yaning Jia, Chunhui Zhang, Xingjian Diao, Xiangchi Yuan, Zhongyu Ouyang, Chiyu Ma, and Soroush Vosoughi. 2025. What makes a good curriculum? disentangling the effects of data ordering on llm mathematical reasoning. *arXiv preprint arXiv:2510.19099*.

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*.

Chiyu Ma, Shuo Yang, Kexin Huang, Jinda Lu, Haoming Meng, Shangshang Wang, Bolin Ding, Soroush Vosoughi, Guoyin Wang, and Jingren Zhou. 2026. Fipo: Eliciting deep reasoning with future-kl influenced policy optimization. *arXiv preprint arXiv:2603.19835*.

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*.

Zhongyu Ouyang, Mingxuan Ju, Soroush Vosoughi, and Yanfang Ye. 2025a. Non-parametric graph convolution for re-ranking in recommendation systems. In *ACM Recommender Systems conference*.

Zhongyu Ouyang, Chunhui Zhang, Shifu Hou, Chuxu Zhang, and Yanfang Ye. 2024. How to improve representation alignment and uniformity in graph-based collaborative filtering? In *The international AAAI conference on web and social media*.

Zhongyu Ouyang, Chunhui Zhang, Yaning Jia, and Soroush Vosoughi. 2025b. Scaled supervision is an implicit lipschitz regularizer. In *The International AAAI Conference on Web and Social Media*.

Tianyu Pang, Yujie Fang, Zihang Liu, Shenyang Deng, Lei Hsiung, Shuhua Yu, and Yaoqing Yang. 2026. Htmuon: Improving muon via heavy-tailed spectral correction. *arXiv preprint arXiv:2603.10067*.

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*.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *The Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*.

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*.

Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *ACM SIGIR conference*.

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*.

Xiangchi Yuan, Xiang Chen, Tong Yu, Dachuan Shi, Can Jin, Wenke Lee, and Saayan Mitra. 2025a. Mitigating forgetting between supervised and reinforcement learning yields stronger reasoners. *arXiv preprint arXiv:2510.04454*.

Xiangchi Yuan, Dachuan Shi, Chunhui Zhang, Zheyuan Liu, Shenglong Yao, Soroush Vosoughi, and Wenke Lee. 2026. Behavior knowledge merge in reinforced agentic models. *arXiv preprint arXiv:2601.13572*.

Xiangchi Yuan, Chunhui Zhang, Zheyuan Liu, Dachuan Shi, Leyan Pan, Soroush Vosoughi, and Wenke Lee. 2025b. Superficial self-improved reasoners benefit from model merging. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

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*.

Chunhui Zhang, Yiren Jian, Zhongyu Ouyang, and Soroush Vosoughi. 2025a. Pretrained image-text models are secretly video captioners. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Chunhui Zhang, Zhongyu Ouyang, Xingjian Diao, Zheyuan Liu, and Soroush Vosoughi. 2025b. Knowing more, acting better: Hierarchical representation for embodied decision-making. In *Findings of the Association for Computational Linguistics: EMNLP 2025*.

Chunhui Zhang, Zhongyu Ouyang, Kwonjoon Lee, Nakul Agarwal, Sean Dae Houlihan, Soroush Vosoughi, and Shao-Yuan Lo. 2025c. Overcoming multi-step complexity in multimodal theory-of-mind reasoning: A scalable bayesian planner. *arXiv preprint arXiv:2506.01301*.

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

Continuing from the main paper, we outline the 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., 2024b), a recent adaptation of DPO designed specifically for sequential recommendation.

Supervised Fine-tuning LMs for Sequential Recommendation. Supervised fine-tuning (Ouyang et al., 2022; Jia et al., 2025) (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 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; Zhang et al., 2025a). 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:

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

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., 2024b) 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:

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

where $\mathcal{T}_d$ contains the item titles of multiple dispreferred items[4].

B Derivation of Preference Distribution

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

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

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

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 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:

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

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

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

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

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

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

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

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 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.

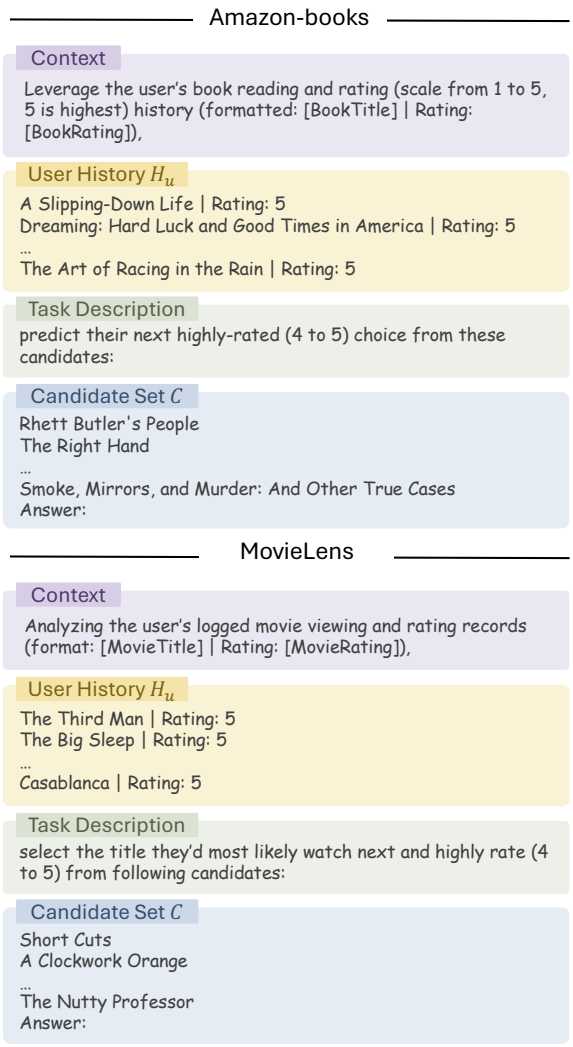

Figure 6: Textual prompt examples for Amazon-books and MovieLens.

C Prompt Examples

We refer the prompts used in previous works (Chen et al., 2024b; Liao et al., 2024) to construct prompts

utilized in our work. Examples in Figure 6 demonstrates 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. 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.

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).

Recent work has also begun to incorporate multiple and fine-grained preference signals into LLM-based recommenders. In particular, ELCoRec (Chen et al., 2024a) integrates numerical and categorical features (e.g., ratings and temporal attributes) with textual representations through feature co-propagation, enhancing the expressiveness of input representations. However, these methods primarily operate at the input or representation level, relying on the model to implicitly infer the relative importance of heterogeneous preference signals during training.

In contrast, our work focuses on preference alignment, explicitly modeling how strongly different preference signals should influence learning by modulating the alignment objective itself. Rather than encoding fine-grained or temporal signals solely as features, we incorporate preference intensity and recency-sensitive temporal context directly into adaptive reward margins, enabling principled control over preference learning dynamics beyond representation enrichment.

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; 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., 2024b) adapts alignment techniques specifically for recommendation tasks, focusing on sequential user preferences and improving the personalization of LLM-based recommenders.

Beyond recommendation, preference alignment has been explored across diverse domains, including logical reasoning (Yuan et al., 2025b,a, 2026; Ma et al., 2026), audio-language reasoning (Diao et al., 2025) and multimodality (Zhang et al., 2025c,b; Diao et al., 2026), demonstrating its broad applicability as a general framework for aligning model behavior with human preferences.

E Experimental Settings

E.1 Datasets

We use five widely used real-world sequential recommendation datasets for evaluation, includ-

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 4: Statistics of datasets

ing *MovieLens-1M*[5] (Harper and Konstan, 2015), *Amazon-books*[6] (Ni et al., 2019), *Steam*[7] (Kang and McAuley, 2018), *BeerAdvocate*[8] (Leskovec and Mcauley, 2012), and *LastFM*[9] (Celma, 2010). We demonstrate the dataset statistics in Table 4. 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 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.

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 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.

[5] https://grouplens.org/datasets/movielens/1m/
[6] https://nijianmo.github.io/amazon/index.html
[7] https://github.com/kang205/SASRec
[8] https://cseweb.ucsd.edu/~jmcauley/datasets.html#multi_aspect
[9] http://ocelma.net/MusicRecommendationDataset/lastfm-1K.html

- 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., 2024b) is a variant of DPO specifically adapted for sequential recommendation that incorporates list-wise negative items in preference alignment.

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., 2024b) 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 § 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 § 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 § 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 Discussion on Potential Risks

This work focuses on methodological limitations and performance gaps, and does not touch ethical, societal, or deployment-related risks such as user manipulation, fairness, or privacy concerns.

G Discussion on Training and Deployment Cost

Our experiments use an 8B-parameter backbone with standard parameter-efficient fine-tuning, which is well within the scale already deployed in many production recommendation stacks; training is a one-time offline cost, while inference reuses the same model for diverse tasks. In addition, RecPO is **orthogonal** to model size, architecture, and optimizer (Pang et al., 2026): the objective can be applied to smaller or specialized backbones, or

combined with distillation/compression, so it does not inherently require heavier deployment than existing LLM-based recommenders.