Objective Shaping with Hard Negatives: Windowed Partial AUC Optimization for RL-based LLM Recommenders

Wentao Shi[1] **Qifan Wang**[1†*] **Chen Chen**[1] **Fei Liu**[1] **Dongfang Liu**[2]
Xu Liu[1] **Wanli Ma**[1] **Junfeng Pan**[1] **Linhong Zhu**[1] **Fuli Feng**[3†]
[1]Meta AI [2]Rochester Institute of Technology
[3]National University of Singapore

Abstract

Reinforcement learning (RL) effectively optimizes Large Language Model (LLM)-based recommenders by contrasting positive and negative items. Empirically, training with beam-search negatives consistently outperforms random negatives, yet the mechanism is not well understood. We address this gap by analyzing the induced optimization objective and show that: (i) Under binary reward feedback, optimizing LLM recommenders with Group Relative Policy Optimization (GRPO) induces an AUC (the Area Under the ROC Curve)-style pairwise surrogate, which is often misaligned with Top-K recommendation; and (ii) Replacing random negatives with beam-search negatives reshapes the objective toward partial AUC, improving alignment with Top-K metrics. Motivated by this perspective, we introduce **W**indowed **P**artial **AUC** (**WPAUC**), which constrains the false positive rate (FPR) to a window $[\alpha, \alpha + d]$ to more directly align with Top-K metrics. We further propose an efficient **T**hreshold-**A**djusted **Win**dowed reweighting (**TAWin**) RL method for its optimization, enabling explicit control over the targeted Top-K performance. Experiments on four real-world datasets validate the theory and deliver consistent state-of-the-art performance. The code and data are available at https://github.com/swt-user/TAWin.

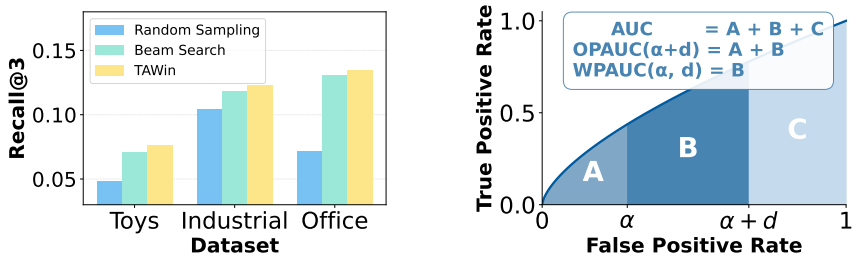
1 Introduction

Recommendation systems are pivotal in mitigating information overload on large-scale platforms by delivering personalized Top-K recommendations [35, 33]. With recent breakthroughs in LLMs [20, 28], LLM-based recommenders have emerged as a promising paradigm for Top-K recommendation [6, 1]. In this setting, RL serves as an effective optimization framework, refining user relative preference modeling by contrasting sampled positive items against negative items [40]. Empirical evidence, including our own reproductions (Figure 1(a)), confirms that RL training using negative items derived from beam search consistently outperforms random sampling [53, 16].

However, the theoretical mechanisms underlying the efficacy of beam search remain underexplored. Existing work generally attributes the advantages of negative items from beam search to their ability to provide more informative training signals [40, 53]. However, such explanations lack the granularity required for algorithmic design. Drawing inspiration from Shi et al. [37], we provide a rigorous theoretical investigation of the optimization objective for RL training, yielding two primary insights:

*† Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Performance Comparison (b) Partial AUC

Figure 1: (a) Performance comparison of Random Sampling, Beam Search ([40]), and our approach (**TAWin**) on three datasets. (b) Illustration of partial AUC on the ROC curve: AUC covers regions $\mathbb{A}+\mathbb{B}+\mathbb{C}$, OPAUC($\alpha+d$) covers $\mathbb{A}+\mathbb{B}$, and WPAUC(α, d) isolates the windowed region $\mathbb{B}$ over the FPR interval $[\alpha, \alpha+d]$.

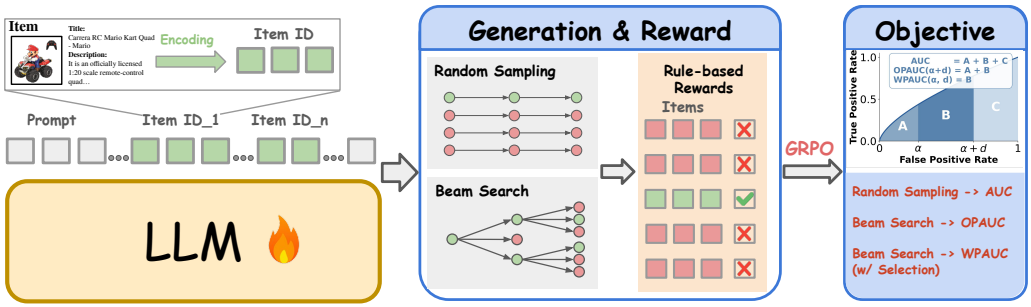


Figure 2: **Overview of GRPO training for LLM-based recommenders.** The LLM recommender generates candidate item IDs from a history prompt via random sampling or beam search, receives rule-based rewards, and is updated with GRPO. Random sampling induces an AUC objective, while beam-search negatives shift the objective toward One-way Partial AUC (OPAUC). Windowed selection over beam-search samples further yields Windowed Partial AUC (WPAUC).

- **Optimizing LLM-based recommenders with GRPO algorithms [36, 12] under binary reward is approximately equivalent to maximizing the AUC metric**. However, AUC is poorly aligned with the Top-K objective in the recommendation task.
- **Incorporating hard negative items generated by beam search into GRPO shifts the objective toward One-way Partial AUC (OPAUC) [10, 26]**, which constrains the false positive rate (FPR) range, as shown in Figure 1(b). **This yields improved alignment with Top-K metrics, following the theoretical analysis in Shi et al. [37].**

Building on these insights, we provide a theoretical explanation for the role of hard negatives in RL-based LLM recommenders: they implicitly steer the optimization objective toward Top-K metrics. However, the correlation between OPAUC and Top-K metrics with specific K remains insufficient, precluding precise control over the optimization process. To address this limitation, we propose a new metric, **W**indowed **P**artial **AUC** (**WPAUC**), which constrains the FPR within a windowed interval $[\alpha, \alpha+d]$, as shown in Figure 1(b). We formally demonstrate that WPAUC exhibits a superior correlation with Top-K metrics and, under specific parameterization, is equivalent to Recall@K. Furthermore, we propose an efficient **T**hreshold-**A**djusted **Win**dowed reweight method (**TAWin**), which emphasizes negative samples within the target window while smoothly attenuating those outside it, thereby preserving the Top-K inductive bias during training. Extensive experiments on four real-world datasets validate our theoretical findings and demonstrate state-of-the-art performance. Our primary contributions are as follows:

- **Theoretical Analysis**: We demonstrate that beam search implicitly steers the RL optimization of LLM recommenders toward Top-K metrics, aligning the objective more closely with recommendation goals.
- **Methodological Innovation**: We propose WPAUC(α, d) metric to establish strong alignment with Top-K metrics and develop **TAWin**, an efficient threshold-adjusted windowed reweighting method, for its optimization.

- **Empirical Validation**: Through extensive testing on four real-world datasets, we confirm our theoretical insights and demonstrate that our approach consistently achieves state-of-the-art results.

2 Preliminary

Task Formulation. We cast LLM-based recommendations as conditional language generation. For each user $u \in \mathcal{U}$ with interaction history $H_u = \{i_1, \ldots, i_n\}$ (items verbalized as text, e.g., titles), the recommender π_θ is trained to generate the **ground-truth target item** i_t under a constrained item vocabulary $\mathcal{I}$ [2, 40]. Formally, the decoding objective is:

$$Y = \mathcal{G}(\pi_\theta, H_u, \mathcal{I}), \quad i = \phi(Y), \tag{1}$$

where $\mathcal{G}$ denotes a constrained generation strategy (e.g., random sampling or beam search), and $\phi(\cdot)$ maps the token sequence $Y = (y_1, \ldots, y_{|Y|})$ to a valid item i.

Constrained Random Sampling sequentially samples tokens from $\pi_\theta(\cdot \mid y_{<j}, H_u)$ while enforcing the valid-token constraint $y_j \in \mathcal{I}$, yielding a stochastic candidate that respects the item vocabulary.

Constrained Beam Search instead performs deterministic decoding by maintaining a beam of the B highest-probability partial hypotheses under the same constraint and returning the B completed sequences (and their mapped items) from the final beam. More implementation details are deferred to Appendix D.

Group Relative Policy Optimization (GRPO). GRPO is a representative reinforcement learning framework for training LLM-based recommenders. At each iteration, it generates a set of G candidate items $\{i_m\}_{m=1}^{G}$ conditioned on each user's history and evaluates them with a verifiable rule-based reward, ensuring alignment with the target item i_t:

$$R_{\text{rule}}(i_m, i_t) = \begin{cases} 1, & \text{if } i_m = i_t, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

GRPO normalizes the rewards within each group to compute token-level advantages. Specifically, for the j-th token in item i_m, the normalized advantage is computed as: $\hat{A}_{m,j} = \frac{r_m - \text{mean}(\{r_m\}_{m=1}^{G})}{\text{std}(\{r_m\}_{m=1}^{G})}$ where r_m denotes rule-based reward $R_{\text{rule}}(i_m, i_t)$, and $\hat{A}_{m,j}$ represents the advantage of j-th token.

The GRPO objective is optimized via a clipped surrogate function (ignoring the KL regularization):

$$\mathcal{J}(\theta) = \mathbb{E}_{H_u, Y \sim \pi_{old}(\cdot|H_u)} \left[\frac{1}{G} \sum_{m=1}^{G} \frac{1}{|Y_m|} \sum_{j=1}^{|Y_m|} \min \left(\rho_{m,j,u} \hat{A}_{m,j}, \text{clip}(\rho_{m,j,u}, 1-\epsilon, 1+\epsilon) \hat{A}_{m,j} \right) \right]. \tag{3}$$

Here, $\rho_{m,j,u} = \frac{\pi_\theta(y_{m,j}|y_{m,<j}, H_u)}{\pi_{\text{old}}(y_{m,j}|y_{m,<j}, H_u)}$, π_{old} is the frozen rollout policy used to sample candidates at each epoch. $\epsilon > 0$ clips per-step updates.

AUC and Partial AUC. In personalized recommendation, AUC quantifies a model's ranking capability by assessing the probability that observed positive items are ranked above unobserved negatives for each user. Formally, it can be expressed in a pairwise ranking perspective as:

$$\text{AUC}_u = \Pr_{i^+ \sim \mathcal{I}_u^+, i^- \sim \mathcal{I}_u^-} \left[f_{u,i^+} > f_{u,i^-} \right], \tag{4}$$

where $\mathcal{I}_u^+$ and $\mathcal{I}_u^-$ are the sets of positive and negative items for user u, respectively, and $f_{u,i}$ represents the predicted score of item i for user u. In LLM-based recommenders, the score is instantiated as the generation probability assigned to the item tokens:

$$f_{u,i} = \prod_{j=1}^{|Y|} \pi_\theta(y_j \mid y_{<j}, H_u), \quad i = \phi(Y). \tag{5}$$

One-way Partial AUC (OPAUC($\alpha + d$)) restricts evaluation to the low-FPR region (FPR $\leq \alpha + d$), thereby emphasizing the ranking quality between positive items and the highest-scored negative items. Formally, the threshold $\eta_{\alpha+d}$ is defined as:

$$\Pr_{i^- \sim \mathcal{I}_u^-} \left[f_{u,i^-} \geq \eta_{\alpha+d} \right] = \alpha + d, \tag{6}$$

and only negative items with prediction scores in $[\eta_{\alpha+d}, 1]$ are included. Shi et al. [37] demonstrate that OPAUC has better alignment to Top-K recommendation performance.

3 Methodology

In this section, we first clarify the role of negative sampling in RL training for LLM-based recommenders (§ 3.1), then propose a new optimization metric WPAUC (§ 3.2), and finally present a smoothed reweight method TAWin (§ 3.3).

3.1 Negative Sampling Meets GRPO

We formalize how the negative sampling distribution in GRPO implicitly specifies its optimization target. Under a binary reward, GRPO can be reformulated as a pairwise objective equivalent to AUC maximization. Moreover, replacing random sampling with beam search shifts negatives toward the high-score tail of $\mathcal{I}_u^-$, inducing an OPAUC objective that better aligns with Top-K ranking metrics, following the analysis of Shi et al. [37].

Analysis of GRPO. We first establish a formal connection between GRPO and pairwise ranking.

Lemma 3.1 (Pairwise Formulation of GRPO)**.** *Under binary reward defined in Eq.* (2) *and constrained random sampling strategy, the GRPO objective $\mathcal{J}(\theta)$ in Eq.* (3) *can be rewritten as*

$$\mathcal{J}(\theta) = \mathbb{E}_{H_u} \sqrt{p(H_u)(1-p(H_u))} \cdot \mathbb{E}_{Y^+ \sim \pi_{old}^+, Y^- \sim \pi_{old}^-} \Big[s_\theta^+(Y^+, H_u) - s_\theta^-(Y^-, H_u) \Big], \quad (7)$$

where $p(H_u) \triangleq \Pr_{Y \sim \pi_{old}(\cdot|H_u)} \big[r(Y \mid H_u) = 1 \big]$, $\pi_{old}^+(\cdot \mid H_u) \triangleq \pi_{old}(\cdot \mid H_u) \mid r(Y \mid H_u) = 1$, $\pi_{old}^-(\cdot \mid H_u) \triangleq \pi_{old}(\cdot \mid H_u) \mid r(Y \mid H_u) = 0$, *and*

$$s_\theta^\pm(Y^\pm, H_u) \triangleq \frac{1}{|Y|} \sum_{j=1}^{|Y|} \begin{cases} \min(\rho_{j,u},\, 1+\epsilon), & (+), \\ \max(\rho_{j,u},\, 1-\epsilon), & (-). \end{cases} \quad (8)$$

where $\rho_{j,u} = \frac{\pi_\theta(y_j | y_{<j}, H_u)}{\pi_{old}(y_j | y_{<j}, H_u)}$.

The proof can be found in Li et al. [18], with a complementary derivation in the Appendix E.

From sequence-level sampling to item-level pairwise ranking. Lemma 3.1 shows that GRPO encourages a higher sequence-level score for the positive response Y^+ than for the negative response Y^-. To relate this sequence-level preference to item-level ranking, we rely on a valid sequence-to-item mapping $i = \phi(Y)$ and an order-consistency condition between the sequence-level score $s_\theta(Y, H_u)$ and the induced item score $f_{u,i}$. Under these conditions, GRPO induces an approximate item-level pairwise ranking signal of the form

$$\Pr_{i^+ \sim \mathcal{I}_u^+,\, i^- \sim \mathbb{P}_{\text{neg}}(\cdot|u)} \left[f_{u,i^+} > f_{u,i^-} \right]. \quad (9)$$

The approximation arises because token-wise clipping in GRPO may break global monotonicity between the sequence-level score and the induced item score. More discussion of the consistency conditions is deferred to Appendix G.1.

From AUC to OPAUC via Beam Search. Compared to AUC, OPAUC($\alpha + d$) restricts attention to top-($\alpha + d$) quantile set of negative items under the policy score $f_{u,i}$:

$$\mathcal{Q}_u^-(\alpha + d) \triangleq \{ i^- \in \mathcal{I}_u^- : f_{u,i^-} \geq \eta_{\alpha+d} \}, \quad (10)$$

where $\eta_{\alpha+d}$ is defined in Eq. (6). This suggests that if the negative sampler is biased toward the upper tail of f_{u,i^-}, GRPO will implicitly optimize a *partial* AUC.

Lemma 3.2 (Constrained Beam Search and Hard-Negative Selection)**.** *In the limit as $B \to \infty$, constrained beam search becomes exactly equivalent to sampling from the top $\eta_{\alpha+d}$ fraction of negatives ranked by $f_{u,i}$.*

The proof can be found in Appendix F. Inspired by the lemma, under constrained beam search with finite beam width B, the induced negative sampler concentrates on $\mathcal{Q}_u^-(\alpha + d)$ and thus approximates hard negative sampling. We can directly obtain the following proposition.

Proposition 3.3 (GRPO with beam negatives optimizes OPAUC)**.** *Consider GRPO training where the positive item is the ground-truth target $i_t \in \mathcal{I}_u^+$ and negatives are generated by constrained beam search. Under Lemma 3.2, the induced pairwise objective in Lemma 3.1 is approximately equivalent to optimizing* $\mathrm{OPAUC}(\alpha + d)$.

The direct derivation can be found in Appendix G.

Implication for Top-K alignment. By concentrating comparisons on high-score negatives, OPAUC exhibits a tighter connection to Top-K ranking metrics than AUC [37]. Therefore, Proposition 3.3 provides a theoretical explanation for why beam-search-based GRPO tends to deliver stronger Top-K performance: beam decoding reshapes the induced negative sampling distribution toward the low-FPR region, yielding a Top-K-aligned partial-AUC objective.

3.2 Windowed Partial AUC (WPAUC)

The above analysis clarifies how beam-search negatives strengthen GRPO by shifting the induced objective toward OPAUC. However, OPAUC only imposes an upper bound on the FPR, making it hard to optimize a specific K. To enable finer-grained control, we introduce WPAUC, which evaluates ranking quality over a shifted low-FPR window and thereby emphasizes a specific Top-K metric.

Definition. Fix a user u with positive set $\mathcal{I}_u^+$ and negative set $\mathcal{I}_u^-$ (with $n^- := |\mathcal{I}_u^-|$). Let $f_{u,i}$ be the predicted score (e.g., the generation probability in Eq. (5)). Let $\{i_{(\sigma)}^-\}_{\sigma=1}^{n^-}$ denote negative items sorted by descending scores, $f_{u,i_{(1)}^-} \geq \cdots \geq f_{u,i_{(n^-)}^-}$. For two FPR bounds $0 \leq \alpha < \alpha + d \leq 1$, we define the *windowed negative set*

$$\mathcal{W}_u^-(\alpha, d) \;\coloneqq\; \Big\{ i_{(\sigma)}^- \in \mathcal{I}_u^- \;:\; \lceil \alpha n^- \rceil < \sigma \leq \lceil (\alpha + d) n^- \rceil \Big\}. \tag{11}$$

We then define $\mathrm{WPAUC}(\alpha, d)$ as:

$$\mathrm{WPAUC}_u(\alpha, d) \;\coloneqq\; \Pr_{i^+ \sim \mathcal{I}_u^+,\, i^- \sim \mathcal{W}_u^-(\alpha,d)} \Big[f_{u,i^+} > f_{u,i^-} \Big]. \tag{12}$$

Notably, OPAUC is a special case: $\mathrm{OPAUC}(\alpha + d) = \mathrm{WPAUC}(0, \alpha + d)$, and AUC corresponds to $\mathrm{AUC} = \mathrm{WPAUC}(0, 1)$.

Why windowing helps Top-K control. Unlike OPAUC, which aggregates all negatives in $[0, \alpha + d]$, WPAUC can *shift* the window to concentrate on the specific FPR region that matches a desired K.

Theorem 3.4 (WPAUC yields tighter Top-K bounds than OPAUC)**.** *Suppose there are n^+ positive items and n^- negative items with $n^+ < K$ and $n^- > K$. Rank all items in descending order according to scores from any model f. Define the FPR window $\alpha = (K - n^+)/n^-, d = n^+/n^-$, then we have that*

$$\Big\lceil n^+ \big(1 - \sqrt{1 - w}\big) \Big\rceil / n^+ \;\leq\; \textit{Recall@}K \;\leq\; \Big\lfloor n^+ \sqrt{w} \Big\rfloor / n^+, \tag{13}$$

where $w =$ WPAUC(α, d). Moreover, for the same K, the interval induced by (13) *is strictly tighter than the corresponding bounds obtained from* $\mathrm{OPAUC}(\alpha + d)$.

A special case: equivalence to Recall@K with one positive. The following lemma formalizes an extreme setting where WPAUC exactly recovers the Recall@K metric.

Lemma 3.5 (One-positive equivalence)**.** *Consider a user with exactly one positive item and n^- negatives. Let $K \geq 2$ and set $\alpha = \frac{K-1}{n^-}, d = \frac{1}{n^-}$. Then*

$$\mathrm{WPAUC}(\alpha, d) \;=\; \mathrm{Recall@K}. \tag{14}$$

The complete proof is deferred to Appendix H.

Empirical Analysis. We provide simulation evidence to validate the Top-K alignment of WPAUC. We conduct Monte Carlo experiments on synthetic rankings: in each trial, we sample a random permutation containing 10 positive items and 200 negative items, and evaluate both $\mathrm{Recall@}K$ and $\mathrm{WPAUC}(\alpha, d)$ over a grid of window parameters. Repeating this procedure for 10,000 Monte Carlo trials, we compute the resulting Pearson correlation between $\mathrm{Recall@}K$ and $\mathrm{WPAUC}(\alpha, d)$ across trials, and report it in Figure 3. From the figure, we observe that:

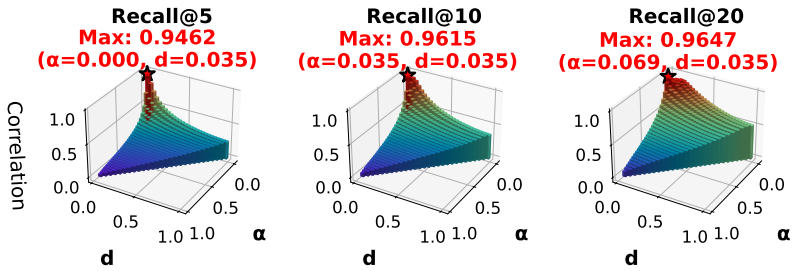


Figure 3: Pearson correlation between Recall@K and WPAUC(α, d) over parameters α and d, for $K \in \{5, 10, 20\}$. The red star marks the maximum correlation setting in each panel.

- For each K, the correlation between WPAUC(α, d) and Recall@K is maximized at a specific (α, d), indicating that an appropriate low-FPR window yields the strongest Top-K alignment.
- The maximizers shift monotonically with K: as K increases, the optimal window moves to larger α (i.e., deeper into the negative-score tail), while the optimal window width d remains relatively small. This trend matches the intuition in Theorem 3.4.

Overall, these results substantiate that optimizing WPAUC(α, d) can serve as a practical surrogate for targeted Top-K ranking quality.

3.3 Threshold-Adjusted Windowed Reweight (TAWin)

While WPAUC motivates windowed negative selection, naively discarding negatives outside the target window is undesirable: it wastes sampled negatives (low sample efficiency, especially for costly RL rollouts) and introduces discontinuous inclusion decisions that amplify gradient variance. To address this, we propose **TAWin**, which replaces hard truncation with a soft threshold-adjusted windowed reweight operator that preserves the inductive bias.

Soft Top-$\hat{K}$ selection operator. To construct TAWin, we first revisit the soft Top-$\hat{K}$ selection operator. Given a score vector $\boldsymbol{x} \in \mathbb{R}^n$, the hard Top-$\hat{K}$ operator $T_{\hat{K}}(\boldsymbol{x})$ returns a $\hat{K}$-hot indicator vector. In contrast, a soft Top-$\hat{K}$ operator defines a continuous mapping $S_{\hat{K},\tau} : \mathbb{R}^n \to \Delta_{\hat{K}}^{n-1}$, where

$$\Delta_{\hat{K}}^{n-1} \triangleq \Big\{ \boldsymbol{p} \in [0,1]^n \;\Big|\; \sum_{i=1}^{n} p_i = \hat{K} \Big\}, \tag{15}$$

where temperature τ controls smoothness. We require: (i) **Monotonicity:** if $x_i \geq x_j$, then $[S_{\hat{K},\tau}(\boldsymbol{x})]_i \geq [S_{\hat{K},\tau}(\boldsymbol{x})]_j$; (ii) **Consistency:** $\lim_{\tau \to 0^+} S_{\hat{K},\tau}(\boldsymbol{x}) = T_{\hat{K}}(\boldsymbol{x})$.

Threshold-adjusted soft Top-$\hat{K}$ operator $\mathcal{T}_{\hat{K},\tau}$. Following Su [39], we use the clipped exponential function $g(x) = \min(1, e^x)$ to instantiate the threshold-adjusted soft Top-$\hat{K}$ selection operator $\mathcal{T}_{\hat{K},\tau}$:

$$\mathcal{T}_{\hat{K},\tau}(\boldsymbol{x}) \triangleq \min\Big(\mathbf{1}, \exp\big(\tfrac{\boldsymbol{x} - \lambda(\boldsymbol{x})\mathbf{1}}{\tau}\big)\Big), \text{s.t. } \sum_{i=1}^{n} \mathcal{T}_{\hat{K},\tau}(\boldsymbol{x})_i = \hat{K}, \tag{16}$$

where $\mathbf{1}$ denotes the n-dimensional all-ones vector and the threshold $\lambda(\boldsymbol{x})$ is the unique scalar that calibrates the total mass to $\hat{K}$. In practice, sort $x_{(1)} \geq \cdots \geq x_{(n)}$ and find an index $m < \hat{K}$ such that $x_{(m)} \geq \lambda \geq x_{(m+1)}$, then $\lambda(\boldsymbol{x})$ has the closed form:

$$\lambda(\boldsymbol{x}) = \tau \log\Big(\sum_{i=m+1}^{n} \exp\big(\tfrac{x_{(i)}}{\tau}\big) \Big) - \tau \log(\hat{K} - m), \tag{17}$$

with m obtained by a short scan over $m \in \{0, \ldots, \hat{K} - 1\}$. As $\tau \to 0^+$, $\mathcal{T}_{\hat{K},\tau}$ becomes sharp and approaches the hard Top-$\hat{K}$ selector $T_{\hat{K}}$.

Threshold-Adjusted Windowed Reweight (TAWin) Then we construct the TAWin method based on the soft Top-$\hat{K}$ operator $\mathcal{T}_{\hat{K},\tau}$. For each user u, constrained beam search yields n candidate

negatives $\mathcal{Y}_u^- = \{Y_1, \ldots, Y_n\}$, which we rank by model scores f. Let $\sigma_u : \mathcal{Y}_u^- \to \{1, \ldots, n\}$ map each candidate Y to its rank $\sigma = \sigma_u(Y)$ (higher score $\Rightarrow$ smaller σ). Define the normalized rank $\tilde{\sigma}(\sigma) \triangleq (\sigma - 1)/(n - 1)$ and an anchor $\sigma_\star$ for the target window's low-FPR boundary. We assign each candidate a logit by its distance to the anchor:

$$x_u(Y) \triangleq -\left|\tilde{\sigma}(\sigma_u(Y)) - \tilde{\sigma}(\sigma_\star)\right|. \tag{18}$$

Applying $\mathcal{T}_{\hat{K},\tau}$ on these logits yields soft windowed weights over the candidates:

$$\boldsymbol{w}_u = \mathcal{T}_{\hat{K},\tau}\big([x_u(Y_1), \ldots, x_u(Y_G)]\big), \tag{19}$$

Here, $\hat{K}$ controls the window width (corresponding to d in WPAUC(α, d)), and anchor $\sigma_\star$ determines the window start (corresponding to α in WPAUC(α, d)). In the sharp limit (small τ), $\boldsymbol{w}_u$ concentrates to a hard window selector $T_{\hat{K}}$.

Based on $\boldsymbol{w}_u$, we define final weight function $\omega_u(\cdot)$ by

$$\omega_u(Y) \triangleq \begin{cases} 1, & Y \in \mathcal{Y}_u^+, \\ \mathrm{Rescale}\big(\boldsymbol{w}_u\big)_{\sigma_u(Y)}, & Y \in \mathcal{Y}_u^-, \end{cases} \tag{20}$$

where $\mathrm{Rescale}(\boldsymbol{w}_u)_\sigma = n * w_{u,\sigma} / \sum_{\sigma'=1}^{n} w_{u,\sigma'}$.

TAWin-Based RL Objective. TAWin performs sequence-level reweighting over rollout candidates by assigning a weight $\omega_u(Y)$ to each sampled sequence Y, and scales its policy-gradient contribution to optimize WPAUC(α, d). This weighting can be applied to multiple RL algorithms (Section 4.4). By default, we use GRPO, yielding:

$$\mathcal{J}_{\text{TAWin}}(\theta) = \mathbb{E}_{H_u,\, Y_m}\left[\frac{1}{n}\sum_{m=1}^{n}\frac{\omega_u(Y_m)}{|Y_m|}\sum_{j=1}^{|Y_m|}\min\left(\rho_{m,j,u}\hat{A}_{m,j},\ \mathrm{clip}(\rho_{m,j,u}, 1-\epsilon, 1+\epsilon)\hat{A}_{m,j}\right)\right], \tag{21}$$

where $\rho_{m,j,u} = \frac{\pi_\theta(y_{m,j}|y_{m,<j},H_u)}{\pi_{\text{old}}(y_{m,j}|y_{m,<j},H_u)}$, and the full algorithm is provided in Appendix I.

4 Experiments

In this section, we evaluate our methods on four public datasets to answer the following questions: **(Q1)** How does TAWin compare with traditional recommenders and LLM recommenders across Top-K metrics? **(Q2)** How does varying TAWin hyperparameters affect performance across different Top-K metrics? **(Q3)** How sensitive are the TAWin to realistic finite beam widths? **(Q4)** How does TAWin generalize across diverse base models, item encoding strategies, and optimization methods?

Datasets and Metrics. We evaluate TAWin on four real-world datasets, including three subsets (**Toys**, **Industrial**, and **Office**) from Amazon Review Dataset[2] [17] and **Yelp** Dataset[3]. To assess Top-K performance, we use Recall (R@K) and Normalized Discounted Cumulative Gain (N@K). More details are provided in Appendix J.1 and J.2.

Baselines. We consider three categories of baseline methods: (1) Traditional sequential recommendation models, including GRU4Rec [13], Caser [41], and SASRec [15]. (2) Generative recommendation models, represented by TIGER [31], MiniOneRec [16]. (3) LLM-based recommendation models, including D^3 [2], S-DPO [7] and ReRe [40]. Additional details are provided in Appendix J.3.

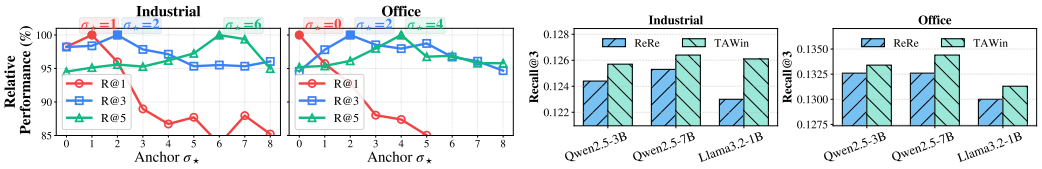
Implementations. Following Tan et al. [40], we train TAWin with a mini-batch size of 512. We use a fixed learning rate of 1×10^{-5} and set the GRPO coefficient β to 1×10^{-3}. For each input, we sample $n = 16$ candidate sequences. We train TAWin for two epochs initialized from the vanilla Qwen2.5-0.5B [29] checkpoint on 8 NVIDIA H800 GPUs. Unless otherwise specified, the parameter K in TAWin is set to 4. The anchor $\sigma_\star$ is selected from $\{0, 1, 2\}$ and the temperature τ is selected via grid search over $\{1/2, 1/3, 1/4, 1/5, 1/6\}$. All LLM-based recommenders use Qwen2.5-0.5B as the backbone. For ReRe, we adopt beam-search generation with rule-based rewards. The ranking reward variant can be viewed as a special reweighting scheme and its detailed analysis and comparisons are provided in Appendix L. Additional baseline hyperparameters are deferred to Appendix J.4.

[2] https://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews

[3] https://business.yelp.com/data/resources/open-dataset/

Table 1: Performance comparison under Top-K metrics across four datasets.

Dataset	Metric	GRU4Rec	Caser	SASRec	TIGER	MiniOneRec	D^3	S-DPO	ReRe	TAWin
Toys	R@1	0.0090	0.0125	0.0216	0.0224	0.0271	0.0371	0.0275	0.0411	**0.0471**
	R@3	0.0169	0.0219	0.0359	0.0383	0.0458	0.0612	0.0534	0.0709	**0.0761**
	N@3	0.0135	0.0180	0.0298	0.0305	0.0378	0.0512	0.0449	0.0583	**0.0639**
Industrial	R@1	0.0461	0.0371	0.0567	0.0632	0.0831	0.0810	0.0635	0.0783	**0.0904**
	R@3	0.0657	0.0624	0.0761	0.0852	0.1100	0.1103	0.1032	0.1184	**0.1237**
	N@3	0.0576	0.0523	0.0682	0.0742	0.0990	0.0980	0.0906	0.1016	**0.1099**
Office	R@1	0.0384	0.0450	0.0641	0.0624	0.0922	0.0810	0.0390	0.0830	**0.0961**
	R@3	0.0631	0.0734	0.0923	0.0986	0.1219	0.1204	0.1169	0.1304	**0.1341**
	N@3	0.0529	0.0614	0.0807	0.0852	0.1137	0.1040	0.1033	0.1115	**0.1187**
Yelp	R@1	0.0074	0.0079	0.0074	0.0068	0.0060	0.0120	0.0189	0.0206	**0.0227**
	R@3	0.0164	0.0217	0.0175	0.0154	0.0163	0.0309	0.0342	0.0360	**0.0370**
	N@3	0.0125	0.0158	0.0133	0.0113	0.0120	0.0228	0.0295	0.0295	**0.0301**



(a) TAWin sensitivity to the anchor $\sigma_\star$.

(b) Comparison across different base models.

Figure 4: Hyperparameter and model-level analyses of TAWin. (a) Sensitivity to the anchor $\sigma_\star$ on two datasets, where the best-performing setting on each curve is highlighted with a filled marker. (b) Comparison between ReRe and TAWin across different base models.

4.1 Overall Experiments (RQ1)

We evaluate the performance of TAWin across four datasets, with the results reported in Table 1. From the table, we have the following key insight:

- **Superiority of TAWin across All Baselines.** TAWin consistently establishes a new state-of-the-art, outperforming all baselines on all datasets. Specifically, TAWin achieves average relative improvements of 84.9%, 52.0%, and 5.5% over traditional, generative, and LLM-based recommenders, respectively. We attribute this performance gain to the windowed integration of negative items within the RL framework, where the training objective aligns better with Top-K metrics.
- **Efficacy of LLM-based Recommenders.** LLM-based recommenders demonstrate a clear margin of superiority over traditional sequential recommenders. This improvement stems from their semantic representations and world knowledge, thus strengthening the modeling of user interests.
- **The Advantage of RL in Alignment.** Our results indicate that RL-based methodologies yield consistent improvements over their Supervised Fine-Tuning (SFT) counterparts. For instance, MiniOneRec surpasses TIGER by an average of 22.6%, while TAWin exceeds BigRec by 49.4%. This is attributed to RL's ability to model users' relative preferences, yielding better alignment with recommendation objectives.

4.2 Effect of Top-K Control in TAWin (RQ2)

To investigate the controllability of the TAWin, we examine whether modulating its hyperparameters can systematically bias optimization toward specific Top-K metrics. Figure 4a illustrates the relative performance across varying anchor parameter ($\sigma_\star$) on the Industrial and Office datasets. Our analysis reveals several key insights:

- **Unimodal Trend as $\sigma_\star$ Varies.** As shown in Figure 4a, for a fixed target R@K with specific K, the relative performance exhibits a unimodal trend as the anchor $\sigma_\star$ varies. Performance reaches its zenith at a specific anchor value, whereas anchor $\sigma_\star$ that are either too small or overly large result in suboptimal alignment with the targeted Top-K metric. This empirical behavior is consistent with our analytical findings in Section 3.2.
- **Optimal $\sigma_\star$ Increases with Target K.** We observe a consistent correlation between the target metric and the optimal anchor value: specifically, smaller target K metrics (e.g., R@1) achieve

Table 2: TAWin sensitivity to finite beam widths on Industrial and Office. Bold marks the best result for each dataset and metric.

Width B	Industrial			Office		
	R@1	R@3	N@3	R@1	R@3	N@3
8	0.0796	0.1134	0.0994	0.0785	0.1200	0.1030
16	0.0891	0.1227	0.1087	**0.0956**	0.1346	0.1187
32	0.0935	**0.1282**	0.1137	**0.0956**	0.1375	**0.1203**
64	**0.0940**	0.1279	**0.1144**	0.0949	**0.1381**	0.1195

Table 3: Comparison of ReRe and TAWin across different optimization algorithms.

		Industrial			Office		
Algorithm	Method	R@1	R@3	N@3	R@1	R@3	N@3
DAPO	ReRe	0.0814	0.1202	0.1039	0.0832	0.1276	0.1098
	TAWin	**0.0875**	**0.1248**	**0.1089**	**0.0910**	**0.1317**	**0.1149**
GSPO	ReRe	0.0736	0.1166	0.0988	0.0816	0.1284	0.1094
	TAWin	**0.0867**	**0.1224**	**0.1073**	**0.0956**	**0.1305**	**0.1163**

peak performance at lower anchor values ($\sigma_\star = 1$ for Industrial, $\sigma_\star = 0$ for Office) compared to larger K metrics like R@5, which peak at higher anchors ($\sigma_\star = 6$ and $\sigma_\star = 4$, respectively). This trend aligns with our theoretical analysis and simulation results in Section 3.2, indicating that TAWin can flexibly adapt its optimization focus toward different Top-K regimes through appropriate hyperparameter control.

4.3 Sensitivity to Finite Beam Width (RQ3)

We evaluate $B \in \{8, 16, 32, 64\}$ on Industrial and Office using 8 NVIDIA H100 GPUs, with $\hat{K} = 4$ and all other hyperparameters unchanged. Table 2 shows that performance generally improves from $B = 8$ to $B = 32$ and then approaches a plateau, with small, nonmonotonic changes at $B = 64$. Smaller beams can prune globally high-scoring negatives, weakening the approximation to upper-tail selection; however, the results indicate that moderate finite beams are sufficient for strong empirical performance. We use $B = 16$ in the main experiments to match the baselines' generation budget. This sensitivity study supports the finite-beam approximation empirically rather than establishing a quantitative bound on its error.

4.4 Generalization Analysis (RQ4)

In this section, we assess the robustness of our theoretical insights and the empirical efficacy of TAWin. We examine TAWin's versatility across diverse backbone models, optimization algorithms, and item-encoding paradigms.

Generalization across Backbone Models. We investigate architectural robustness by evaluating models from distinct families and varying scales: Qwen2.5 [29] (3B and 7B) and Llama-3.2 [11] (1B). Figure 4b compares the performance of ReRe and TAWin under these configurations. TAWin consistently outperforms ReRe across all backbones, indicating that the benefits of reshaping the RL objective via hard-negative sampling are model-agnostic. Furthermore, the performance gains remain stable as model size increases, suggesting that TAWin possesses favorable scalability.

Generalization across Optimization Algorithms. To evaluate generalization across RL optimization algorithms, we extend our evaluation to GSPO [51] and DAPO [45]. Given that analogous AUC-style reformulations for these algorithms are established in Li et al. [18], we investigate whether TAWin's negative-sampling strategy enhances alignment with Top-K recommendation metrics. Table 3 presents the performance on the Toys, Industrial, and Office datasets. TAWin consistently surpasses ReRe under both optimization algorithms, demonstrating that the method's improvements are transferable across different RL optimizers.

Table 4: Performance comparison based on semantic-ID item encoding strategies.

Method	Industrial			Office		
	R@1	R@3	N@3	R@1	R@3	N@3
MiniOneRec-SFT	0.0726	0.0986	0.0877	0.0900	0.1196	0.1074
MiniOneRec-GRPO	0.0831	0.1100	0.0989	0.0969	0.1325	0.1174
MiniOneRec-TAWin	**0.0862**	**0.1158**	**0.1033**	**0.0974**	**0.1378**	**0.1253**

Generalization across Item Encoding Strategies. Finally, we evaluate generalization beyond title-based encoding by considering semantic item representations, specifically utilizing MiniOneRec [16]. We integrate TAWin's negative sampling strategy into the MiniOneRec training process (denoted as MiniOneRec-TAWin) and compare it against the standard MiniOneRec baseline. As shown in Table 4, MiniOneRec-TAWin yields consistent improvements on both the Industrial and Office datasets, validating the robustness of TAWin irrespective of the underlying item-encoding strategy.

5 Conclusion

In this work, we provide a principled explanation of why hard negatives benefit RL-trained LLM recommenders: GRPO with binary rewards optimizes AUC, while beam-search negatives reshape the objective toward Partial AUC, improving alignment with Top-K metrics. Building on this insight, we propose WPAUC for controllable Top-K–oriented optimization and TAWin for efficient soft window selection. Extensive experiments across datasets, backbones, item encodings, and RL optimizers validate the theory and show consistent state-of-the-art gains.

Acknowledgments and Disclosure of Funding

We gratefully acknowledge Meta for providing the computational resources that supported the model training experiments.

References

[1] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *RecSys*, pages 1007–1014. ACM, 2023.

[2] Keqin Bao, Jizhi Zhang, Yang Zhang, Xinyue Huo, Chong Chen, and Fuli Feng. Decoding matters: Addressing amplification bias and homogeneity issue for llm-based recommendation. *arXiv preprint arXiv:2406.14900*, 2024.

[3] Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yanchen Luo, Chong Chen, Fuli Feng, and Qi Tian. A bi-step grounding paradigm for large language models in recommendation systems. *Trans. Recomm. Syst.*, 3(4):53:1–53:27, 2025.

[4] Hugo Caselles-Dupré, Florian Lesaint, and Jimena Royo-Letelier. Word2vec applied to recommendation: hyperparameters matter. In *RecSys*, pages 352–356. ACM, 2018.

[5] Jin Chen, Defu Lian, Binbin Jin, Kai Zheng, and Enhong Chen. Learning recommenders for implicit feedback with importance resampling. In *WWW*, pages 1997–2005. ACM, 2022.

[6] Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. On sampling strategies for neural network-based collaborative filtering. In *KDD*, pages 767–776. ACM, 2017.

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

[8] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025.

[9] Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *CoRR*, abs/2502.18965, 2025.

[10] Yuan Gao, Xiang Wang, Xiangnan He, Huamin Feng, and Yong-Dong Zhang. Rumor detection with self-supervised learning on texts and social graph. *Frontiers Comput. Sci.*, 17(4):174611, 2023.

[11] Aaron Grattafiori, Abhimanyu Dubey, and et al. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

[12] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[14] Yupeng Hou, Zhankui He, Julian J. McAuley, and Wayne Xin Zhao. Learning vector-quantized item representation for transferable sequential recommenders. In *WWW*, pages 1162–1171. ACM, 2023.

[15] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.

[16] Xiaoyu Kong, Leheng Sheng, Junfei Tan, Yuxin Chen, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. Minionerec: An open-source framework for scaling generative recommendation. *arXiv preprint arXiv:2510.24431*, 2025.

[17] Himabindu Lakkaraju, Julian McAuley, and Jure Leskovec. What's in a name? understanding the interplay between titles, content, and communities in social media. In *Proceedings of the international AAAI conference on web and social media*, volume 7, pages 311–320, 2013.

[18] Gang Li, Ming Lin, Tomer Galanti, Zhengzhong Tu, and Tianbao Yang. Disco: Reinforcing large reasoning models with discriminative constrained optimization. *arXiv preprint arXiv:2505.12366*, 2025.

[19] Defu Lian, Qi Liu, and Enhong Chen. Personalized ranking with importance sampling. In *WWW*, pages 1093–1103. ACM / IW3C2, 2020.

[20] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[21] Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. Generative recommender with end-to-end learnable item tokenization. In *SIGIR*, pages 729–739. ACM, 2025.

[22] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. Is chatgpt a good recommender? A preliminary study. *CoRR*, abs/2304.10149, 2023.

[23] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017.

[24] Haokai Ma, Ruobing Xie, Lei Meng, Fuli Feng, Xiaoyu Du, Xingwu Sun, Zhanhui Kang, and Xiangxu Meng. Negative sampling in recommendation: A survey and future directions. *CoRR*, abs/2409.07237, 2024.

[25] Donna McClish. Analyzing a portion of the roc curve. *Medical decision making : an international journal of the Society for Medical Decision Making*, 9:190–5, 08 1989. doi: 10.1177/0272989X8900900307.

[26] Harikrishna Narasimhan and Shivani Agarwal. $\text{Svm}_{\text{pauc}}^{\text{tight}}$: a new support vector method for optimizing partial AUC based on a tight convex upper bound. In *KDD*, pages 167–175. ACM, 2013.

[27] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

[28] OpenAI. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[29] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[30] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019.

[31] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Mahesh Sathiamoorthy. Recommender systems with generative retrieval. In *NeurIPS*, 2023.

[32] Steffen Rendle and Christoph Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *WSDM*, pages 273–282. ACM, 2014.

[33] Steffen Rendle and Christoph Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 273–282, 2014.

[34] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461. AUAI Press, 2009.

[35] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[36] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

[37] Wentao Shi, Jiawei Chen, Fuli Feng, Jizhi Zhang, Junkang Wu, Chongming Gao, and Xiangnan He. On the theories behind hard negative sampling for recommendation. In *WWW*, pages 812–822. ACM, 2023.

[38] Wentao Shi, Chenxu Wang, Fuli Feng, Yang Zhang, Wenjie Wang, Junkang Wu, and Xiangnan He. Lower-left partial auc: An effective and efficient optimization metric for recommendation. In *Proceedings of the ACM Web Conference 2024*, pages 3253–3264, 2024.

[39] Jianlin Su. Backpropagating through softmax: Seeking a smooth approximation to top-k, September 2024. URL https://www.spaces.ac.cn/archives/10373.

[40] Junfei Tan, Yuxin Chen, An Zhang, Junguang Jiang, Bin Liu, Ziru Xu, Zhu Han, Jian Xu, Bo Zheng, and Xiang Wang. Reinforced preference optimization for recommendation. *CoRR*, abs/2510.12211, 2025.

[41] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 565–573, 2018.

[42] Hanfang Yang, Kun Lu, Xiang Lyu, and Feifang Hu. Two-way partial auc and its properties. *Statistical methods in medical research*, 28(1):184–195, 2019.

[43] Zhiyong Yang, Qianqian Xu, Shilong Bao, Yuan He, Xiaochun Cao, and Qingming Huang. When all we need is a piece of the pie: A generic framework for optimizing two-way partial auc. In *International Conference on Machine Learning*, pages 11820–11829. PMLR, 2021.

[44] Zhiyong Yang, Qianqian Xu, Shilong Bao, Yuan He, Xiaochun Cao, and Qingming Huang. Optimizing two-way partial auc with an end-to-end framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10228–10246, 2022.

[45] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[46] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, Yinghai Lu, and Yu Shi. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. In *ICML*. OpenReview.net, 2024.

[47] Junjie Zhang, Ruobing Xie, Yupeng Hou, Xin Zhao, Leyu Lin, and Ji-Rong Wen. Recommendation as instruction following: A large language model empowered recommendation approach. *ACM Trans. Inf. Syst.*, 43(5):114:1–114:37, 2025.

[48] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *SIGIR*, pages 785–788. ACM, 2013.

[49] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation. In *ICDE*, pages 1435–1448. IEEE, 2024.

[50] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1435–1448. IEEE, 2024.

[51] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

[52] Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Shiyao Wang, Weifeng Ding, Wuchao Li, Xinchen Luo, Xingmei Wang, Zexuan Cheng, Zixing Zhang, Bin Zhang, Boxuan Wang, Chaoyi Ma, Chengru Song, Chenhui Wang, Di Wang, Dongxue Meng, Fan Yang, Fangyu Zhang, Feng Jiang, Fuxing Zhang, Gang Wang, Guowang Zhang, Han Li, Hengrui Hu, Hezheng Lin, Hongtao Cheng, Hongyang Cao, Huanjie Wang, Jiaming Huang, Jiapeng Chen, Jiaqiang Liu, Jinghui Jia, Kun Gai, Lantao Hu, Liang Zeng, Liao Yu, Qiang Wang, Qidong Zhou, Shengzhe Wang, Shihui He, Shuang Yang, Shujie Yang, Sui Huang, Tao Wu, Tiantian He, Tingting Gao, Wei Yuan, Xiao Liang, Xiaoxiao Xu, Xugang Liu, Yan Wang, Yi Wang, Yiwu Liu, Yue Song, Yufei Zhang, Yunfan Wu, Yunfeng Zhao, and Zhanyu Liu. Onerec technical report. *CoRR*, abs/2506.13695, 2025.

[53] Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Shiyao Wang, et al. Onerec technical report. *arXiv preprint arXiv:2506.13695*, 2025.

[54] Guorui Zhou, Hengrui Hu, Hongtao Cheng, Huanjie Wang, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Lu Ren, Liao Yu, Pengfei Zheng, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Ruiming Tang, Shiyao Wang, Shujie Yang, Tao Wu, Wuchao Li, Xinchen Luo, Xingmei Wang, Yi Su, Yunfan Wu, Zexuan Cheng, Zhanyu Liu, Zixing Zhang, Bin Zhang, Boxuan Wang, Chaoyi Ma, Chengru Song, Chenhui Wang, Chenglong Chu, Di Wang, Dongxue Meng, Dunju Zang, Fan Yang, Fangyu Zhang, Feng Jiang, Fuxing Zhang, Gang Wang, Guowang Zhang, Han Li, Honghui Bao, Hongyang Cao, Jiaming Huang, Jiapeng Chen, Jiaqiang Liu, Jinghui Jia, Kun Gai, Lantao Hu, Liang Zeng, Qiang Wang, Qidong Zhou, Rongzhou Zhang, Shengzhe Wang, Shihui He, Shuang Yang, Siyang Mao, Sui Huang, Tiantian He, Tingting Gao, Wei Yuan, Xiao Liang, Xiaoxiao Xu, Xugang Liu, Yan Wang, Yang Zhou, Yi Wang, Yiwu Liu, Yue Song, Yufei Zhang, Yunfeng Zhao, Zhixin Ling, and Ziming Li. Onerec-v2 technical report. *CoRR*, abs/2508.20900, 2025.

A Limitation

Our theoretical results should be viewed as a principled asymptotic characterization rather than a strict equivalence in full generality. The connection from sequence-level optimization to item-level AUC relies on order consistency between the induced item score and the sequence-level score. While this condition is exact for vanilla policy gradient, token-wise clipping in GRPO breaks global monotonicity, making the induced connection to item-level AUC approximate rather than exact. The analysis of beam-search negatives is also exact only in the infinite-beam-width limit. With a finite beam width, beam search no longer yields exact truncation, but still induces an approximate upper-tail negative set, and therefore serves as a principled approximation to the corresponding low-FPR window. These assumptions allow us to expose the objective-level effect of hard negatives in a clean and analyzable form, but they also define the scope within which the theoretical claims should be interpreted. Finally, our objective-shaping view is not meant to be an exhaustive explanation of the empirical gains from beam-search negatives. A more complete analysis of finite-beam behavior and its interaction with other training dynamics is left for future work.

B Broader Impact

This work contributes to a deeper understanding of objective design in reinforcement learning–based LLM recommender systems by revealing how negative sampling implicitly reshapes optimization targets. By formalizing the connection between hard negatives, Partial AUC, and Top-K metrics, our analysis provides a principled foundation for developing recommendation algorithms that are better aligned with practical evaluation criteria, potentially leading to more effective and reliable personalization systems.

From an applied perspective, the proposed WPAUC metric and TAWin optimization framework offer practitioners interpretable and controllable tools for tailoring recommendation behavior to specific Top-K regimes, which may improve user satisfaction and system efficiency in large-scale online platforms. Beyond recommendations, our insights into objective shaping via sampling strategies may inform the design of RL objectives in other LLM-based decision-making and ranking tasks.

As with all recommendation technologies, improved personalization may amplify existing issues such as filter bubbles or unequal exposure if deployed without appropriate safeguards. We encourage future work to combine objective-aligned optimization with fairness, diversity, and transparency considerations.

C Relation Work

Generative Recommendation Driven by rapid advances in large generative models [27, 8], recommendation is shifting toward generative modeling [1, 46], where models directly generate item identifiers. Two main paradigms have emerged: semantic-ID generation with dense item codes [31, 14, 49] and LLM-based recommendation framed as natural language generation [22, 47].

Aligning these generative models with user interests is central. Supervised fine-tuning [21, 3] is limited by likelihood-based objectives that weakly capture relative preferences, motivating RL-based tuning [7, 40]. The OneRec series [9, 52, 54] shows that SFT followed by RL yields substantial gains, and recent studies further report consistent benefits from beam-search sampling during RL [40, 52]—but the mechanism is unclear. We address this gap with a unified theoretical framework explaining why beam-search sampling improves RL-based tuning.

Negative Sampling for Recommendation In discriminative recommender systems, negative sampling is essential for scalable training [24]. Existing methods fall into static and dynamic schemes: static samplers (e.g., uniform or popularity-based) draw negatives from fixed item distributions [34, 4, 6], whereas dynamic samplers (e.g., DNS [48], PRIS [19], AdaSIR [5]) leverage model scores to prioritize hard negatives [32]. Prior theory Shi et al. [37] shows that, under supervised learning, negative sampling can implicitly steer optimization toward Top-K metrics and improve ranking. Whether this mechanism extends to RL-based generative recommendations remains unclear.

Partial AUC Optimization Partial AUC, introduced in McClish [25], evaluates ROC performance over restricted FPR/TPR ranges. OPAUC targets low-FPR regimes relevant to drug discovery [10, 26], and its connection to Top-K recommendation has been analyzed in Shi et al. [37]. Subsequent variants include TPAUC [42–44], which focuses on the upper-left ROC region but aligns weakly with ranking metrics, and Low-Left Partial AUC [38], which improves Top-K alignment yet suffers from unstable optimization. Motivated by these gaps, we propose WPAUC, a simple and practical objective that more directly aligns with Top-K ranking goals.

D Generation Strategy

We provide the formal definitions of the two constrained generation strategies used in $Y = \mathcal{G}(\pi_\theta, H_u, \mathcal{I})$.

Tokenization and item constraint. Let $\mathcal{V}$ be the base token vocabulary of the LLM, and let $\mathrm{eos} \in \mathcal{V}$ be the end-of-sequence token. Each valid item $i \in \mathcal{I}$ is verbalized into a unique token sequence via a deterministic serialization function $\tau : \mathcal{I} \to \mathcal{V}^+$, where $\tau(i) = (y_1, \ldots, y_{L_i})$. We interpret the constrained item vocabulary $\mathcal{I}$ as a *prefix constraint* over token sequences: a partial hypothesis $y_{1:j} = (y_1, \ldots, y_j)$ is feasible if it is a prefix of at least one serialized item.

Formally, define the set of *admissible next tokens* after a prefix $y_{<j} = (y_1, \ldots, y_{j-1})$ as

$$\mathcal{A}(y_{<j}; \mathcal{I}) \;=\; \Big\{ y \in \mathcal{V} \;\Big|\; \exists\, i \in \mathcal{I} \text{ s.t. } \tau(i) \text{ has prefix } (y_{<j}, y) \Big\}. \tag{22}$$

In practice, $\mathcal{A}(\cdot; \mathcal{I})$ can be implemented efficiently with a prefix trie built over $\{\tau(i) : i \in \mathcal{I}\}$.

The mapping $\phi(\cdot)$ in the main text is then defined by exact match:

$$\phi(Y) = i \quad \Longleftrightarrow \quad \tau(i) = Y, \tag{23}$$

which is well-defined when serialization is unique.

Masked (constrained) token distribution. Given the model π_θ, history H_u, and prefix $y_{<j}$, define the constrained distribution over the next token by masking invalid tokens and renormalizing:

$$\tilde{\pi}_\theta(y \mid y_{<j}, H_u, \mathcal{I}) \;=\; \frac{\pi_\theta(y \mid y_{<j}, H_u) \cdot \mathbf{1}[y \in \mathcal{A}(y_{<j}; \mathcal{I})]}{\sum_{y' \in \mathcal{A}(y_{<j}; \mathcal{I})} \pi_\theta(y' \mid y_{<j}, H_u)}. \tag{24}$$

When $\mathcal{A}(y_{<j}; \mathcal{I})$ is constructed from a trie over $\{\tau(i)\}$, the denominator is guaranteed to be nonzero for any feasible prefix.

D.1 Constrained Random Sampling

Definition. Constrained random sampling generates a stochastic sequence by sampling each token sequentially from the masked distribution:

$$y_j \;\sim\; \tilde{\pi}_\theta(\cdot \mid y_{<j}, H_u, \mathcal{I}), \qquad j = 1, 2, \ldots, T, \tag{25}$$

$$Y \;=\; (y_1, \ldots, y_T), \qquad \text{where } T = \min\{j : y_j = \mathrm{eos}\} \text{ (or a max length cutoff).} \tag{26}$$

Equivalently, the induced sampling distribution over full sequences is

$$\Pr_{\mathcal{G}_{\mathrm{RS}}}(Y \mid H_u, \mathcal{I}) \;=\; \prod_{j=1}^{|Y|} \tilde{\pi}_\theta(y_j \mid y_{<j}, H_u, \mathcal{I}), \tag{27}$$

and the decoded item is $i = \phi(Y)$.

Output. For single-item recommendation, $\mathcal{G}_{\mathrm{RS}}$ returns one Y (and thus one item $\phi(Y)$). If multiple candidates are needed (e.g., for negative sampling), repeat Eq. (25) independently to obtain a multiset of samples.

Algorithm 1 Constrained Beam Search $\mathcal{G}_{\mathrm{BS}}(\pi_\theta, H_u, \mathcal{I})$

Input: model π_θ, history H_u, item set $\mathcal{I}$ (via trie), beam width B, return size B, max length $L_{\max}$
Output: Top-B decoded sequences $\{Y^{(b)}\}_{b=1}^{B}$ and mapped items $\{i^{(b)} = \phi(Y^{(b)})\}_{b=1}^{B}$
Initialize beam $\mathcal{B}_0 \leftarrow \{()\}$ and completed set $\mathcal{F} \leftarrow \emptyset$
for $j = 1$ **to** $L_{\max}$ **do**
  Initialize candidate set $\mathcal{C}_j \leftarrow \emptyset$
  for all $y_{<j} \in \mathcal{B}_{j-1}$ **do**
    Compute admissible token set $\mathcal{A}(y_{<j}; \mathcal{I})$ (Eq. (22))
    for all $v \in \mathcal{A}(y_{<j}; \mathcal{I})$ **do**
      Add candidate $(y_{<j}, y)$ into $\mathcal{C}_j$
    end for
  end for
  Set $\mathcal{B}_j \leftarrow \mathrm{TopB}(\mathcal{C}_j;\ h_\theta(\cdot\,;\, H_u))$ using the score in Eq. (28)
  for all $y \in \mathcal{B}_j$ **do**
    if $y_{|y|} = \mathrm{eos}$ **and** $\exists\, i \in \mathcal{I}$ s.t. $\tau(i) = Y$ **then**
      $\mathcal{F} \leftarrow \mathcal{F} \cup \{Y\}$
    end if
  end for
  if the stopping criterion is satisfied **then**
    break
  end if
end for
Return $\{Y^{(b)}\}_{b=1}^{B} = \mathrm{TopB}(\mathcal{F};\ h_\theta(\cdot\,;\, H_u))$ and $i^{(b)} = \phi(Y^{(b)})$ for $b = 1, \ldots, B$

D.2 Constrained Beam Search

Prefix scoring. For any partial hypothesis $y_{1:j}$, define its cumulative log-likelihood score under π_θ as

$$h_\theta(y_{1:j}\,;\, H_u) \;=\; \sum_{t=1}^{j} \log \pi_\theta(y_t \mid y_{<t}, H_u). \tag{28}$$

Beam update under the constraint. Let B be the beam width. Beam search maintains a set of active prefixes $\mathcal{B}_j$ after j decoding steps. Initialize $\mathcal{B}_0 = \{()\}$ with the empty prefix and an empty set of completed hypotheses $\mathcal{F} = \emptyset$. At step $j \geq 1$, expand each active prefix $y_{<j} \in \mathcal{B}_{j-1}$ only with admissible tokens:

$$\mathcal{C}_j = \bigcup_{y_{<j} \in \mathcal{B}_{j-1}} \Big\{ (y_{<j}, y) \;:\; v \in \mathcal{A}(y_{<j}; \mathcal{I}) \Big\}. \tag{29}$$

Select the top-B candidates by score to form the next beam:

$$\mathcal{B}_j = \mathrm{TopB}\left(\mathcal{C}_j;\ h_\theta(\cdot\,;\, H_u)\right), \tag{30}$$

where $\mathrm{TopB}(\cdot)$ returns the B highest-scoring sequences. Any hypothesis that ends with eos and exactly matches a valid item serialization is moved to the completed set:

$$\mathcal{F} \leftarrow \mathcal{F} \;\cup\; \{y \in \mathcal{B}_j : y_{|Y|} = \mathrm{eos} \;\wedge\; \exists\, i \in \mathcal{I} \text{ s.t. } \tau(i) = Y\}. \tag{31}$$

Decoding stops when either (i) a maximum length $L_{\max}$ is reached, or (ii) all beams are completed.

Returning Top-B sequences and mapped items. Let $\mathrm{TopB}(\mathcal{F})$ denote the top-B completed sequences in $\mathcal{F}$ by $h_\theta(\cdot\,;\, H_u)$. Constrained beam search returns

$$\{Y^{(1)}, \ldots, Y^{(B)}\} = \mathrm{TopB}\left(\mathcal{F};\ h_\theta(\cdot\,;\, H_u)\right), \qquad i^{(b)} = \phi\Big(Y^{(b)}\Big). \tag{32}$$

The algorithm can be found in Algorithm 1.

E Proof of Lemma 3.1

Proof. We focus on the reward-driven term in Eq. (3) and omit the KL regularizer since it does not depend on the binary reward outcomes. Under constrained random sampling, the G rollouts $\{Y_m\}_{m=1}^{G}$ are i.i.d. from $\pi_{old}(\cdot \mid H_u)$, hence by symmetry the outer averaging over m satisfies

$$\mathbb{E}_{\{Y_m\}\sim\pi_{old}(\cdot|H_u)}\Big[\frac{1}{G}\sum_{m=1}^{G} F(Y_m)\Big] = \mathbb{E}_{Y\sim\pi_{old}(\cdot|H_u)}[F(Y)]. \tag{33}$$

Therefore, it suffices to analyze a generic rollout $Y \sim \pi_{old}(\cdot \mid H_u)$.

Let the binary reward for a rollout be

$$r(Y \mid H_u) \;\triangleq\; R_{\text{rule}}(\phi(Y), i_t) \in \{0, 1\}. \tag{34}$$

Define

$$p(H_u) \;\triangleq\; \mathbb{E}_{Y\sim\pi_{old}(\cdot|H_u)}[r(Y \mid H_u)] = \Pr_{Y\sim\pi_{old}(\cdot|H_u)}\big[r(Y \mid H_u) = 1\big], \tag{35}$$

so that $\mathrm{Var}(r(Y \mid H_u)) = p(H_u)\big(1 - p(H_u)\big)$. Following the standard population-normalization analysis (which is exact when the empirical mean/std are replaced by their expectations, and becomes accurate as G grows), the normalized advantage can be written as

$$A(Y \mid H_u) \;\triangleq\; \frac{r(Y \mid H_u) - p(H_u)}{\sqrt{p(H_u)\big(1 - p(H_u)\big)}} = \begin{cases} \sqrt{\dfrac{1 - p(H_u)}{p(H_u)}}, & r(Y \mid H_u) = 1, \\ -\sqrt{\dfrac{p(H_u)}{1 - p(H_u)}}, & r(Y \mid H_u) = 0. \end{cases} \tag{36}$$

Moreover, since $r(Y \mid H_u)$ is item-level, the same advantage applies to every token in the sequence, i.e., $\hat{A}_{m,j} = A(Y_m \mid H_u)$.

Now consider the clipped surrogate inside Eq. (3). For a token ratio

$$x_j(Y, H_u) \;\triangleq\; \frac{\pi_\theta(y_j \mid y_{<j}, H_u)}{\pi_{\text{old}}(y_j \mid y_{<j}, H_u)}, \tag{37}$$

the GRPO token contribution is

$$\min\Big[x_j(Y, H_u)\, A(Y \mid H_u),\; \mathrm{clip}(x_j(Y, H_u), 1 - \epsilon, 1 + \epsilon)\, A(Y \mid H_u)\Big]. \tag{38}$$

When $A(Y \mid H_u) > 0$, the above equals

$$A(Y \mid H_u) \cdot \min\big(x_j(Y, H_u), 1 + \epsilon\big) \;=\; A(Y \mid H_u) \cdot g^{+}\big(x_j(Y, H_u), 1\big), \tag{39}$$

and when $A(Y \mid H_u) \le 0$, it equals

$$-|A(Y \mid H_u)| \cdot \max\big(x_j(Y, H_u), 1 - \epsilon\big) \;=\; -|A(Y \mid H_u)| \cdot g^{-}\big(x_j(Y, H_u), 1\big), \tag{40}$$

where $g^{+}(x, 1) = \min(x, 1 + \epsilon)$ and $g^{-}(x, 1) = \max(x, 1 - \epsilon)$.

Putting these together, the reward-driven objective becomes

$$\mathcal{J}(\theta) = \mathbb{E}_{H_u}\mathbb{E}_{Y\sim\pi_{old}(\cdot|H_u)}\Bigg[\frac{1}{|Y|}\sum_{j=1}^{|Y|}\Big(\mathbb{I}\{r(Y \mid H_u) = 1\}A(Y \mid H_u)\, g^{+}(x_j(Y, H_u), 1) - \mathbb{I}\{r(Y \mid H_u) = 0\}|A(Y \mid H_u)|\, g^{-}(x_j(Y, H_u), 1)\Big)\Bigg]. \tag{41}$$

Conditioning on the two reward events and using (36), we obtain

$$\begin{aligned}\mathcal{J}(\theta) &= \mathbb{E}_{H_u}\Bigg[p(H_u)\cdot\sqrt{\frac{1-p(H_u)}{p(H_u)}}\,\mathbb{E}_{Y^+\sim\pi^+_{old}(\cdot|H_u)}\frac{1}{|Y^+|}\sum_{j=1}^{|Y^+|} g^+\big(x_j(Y^+,H_u),1\big)\\ &\qquad -(1-p(H_u))\cdot\sqrt{\frac{p(H_u)}{1-p(H_u)}}\,\mathbb{E}_{Y^-\sim\pi^-_{old}(\cdot|H_u)}\frac{1}{|Y^-|}\sum_{j=1}^{|Y^-|} g^-\big(x_j(Y^-,H_u),1\big)\Bigg]\\ &= \mathbb{E}_{H_u}\sqrt{p(H_u)\big(1-p(H_u)\big)}\Big[\mathbb{E}_{Y^+\sim\pi^+_{old}(\cdot|H_u)}s^+_\theta(Y^+,H_u)-\mathbb{E}_{Y^-\sim\pi^-_{old}(\cdot|H_u)}s^-_\theta(Y^-,H_u)\Big]\\ &= \mathbb{E}_{H_u}\sqrt{p(H_u)\big(1-p(H_u)\big)}\;\mathbb{E}_{Y^+\sim\pi^+_{old}(\cdot|H_u),\,Y^-\sim\pi^-_{old}(\cdot|H_u)}\big[s^+_\theta(Y^+,H_u)-s^-_\theta(Y^-,H_u)\big],\end{aligned} \tag{42}$$

which is exactly Eq. (7). This completes the proof. □

F Proof of Lemma 3.2

Proof. Fix a user history H_u and consider constrained decoding over the finite item vocabulary $\mathcal{I}$, where each valid item $i \in \mathcal{I}$ corresponds to a unique constrained token sequence Y with $\phi(Y) = i$. Under the policy $\pi_\theta(\cdot \mid H_u)$, the (policy) score of item i is

$$f_{u,i} \triangleq \prod_{j=1}^{|Y(i)|} \pi_\theta(y_j(i) \mid y_{<j}(i), H_u)\,, \tag{43}$$

which is exactly the probability of generating $Y(i)$ under constrained generation.

Constrained beam search with beam size B keeps, at each decoding step, the B highest-probability partial hypotheses (prefixes) under $\pi_{\text{old}}(\cdot \mid H_u)$, and finally returns the Top-K completed sequences in the terminal beam. Because the constrained item vocabulary is finite, the set of valid completed sequences

$$\mathcal{Y}(H_u) \triangleq \{Y : \phi(Y) = i, i \in \mathcal{I}\} \tag{44}$$

is finite. Let N_j denote the number of valid partial prefixes of length j consistent with the constraint. Then for any fixed depth j, if $B \geq N_j$, beam search retains *all* valid prefixes of length j (since there are at most N_j such prefixes). Consequently, in the limit $B \to \infty$, for every depth j the beam contains all valid prefixes, and thus at termination the search enumerates all valid completed sequences in $\mathcal{Y}(H_u)$ and ranks them exactly by their probabilities.

Let $\mathcal{I}_u^-$ be the negative-item set for user u, with $n_u^- \triangleq |\mathcal{I}_u^-|$, and sort negatives by score in descending order:

$$f_{u,i^-_{(1)}} > f_{u,i^-_{(2)}} > \cdots > f_{u,i^-_{(n_u^-)}}\,, \tag{45}$$

In the limit $B \to \infty$, the Top-K completed sequences returned by constrained beam search correspond exactly to the Top-K items

$$\{i^-_{(1)}, \ldots, i^-_{(K)}\} \quad \text{ranked by } f_{u,i}. \tag{46}$$

Now set $\alpha + d \triangleq K/n_u^-$ and let $\eta_{\alpha+d}$ be the OPAUC threshold defined by Eq. (6), i.e., $\Pr_{i^-\sim\mathcal{I}_u^-}\big[f_{u,i^-} \geq \eta_{\alpha+d}\big] = \alpha + d$. Under the no-tie assumption (or after tie breaking), this implies

$$\mathcal{Q}_u^-(\alpha+d) = \{i^- \in \mathcal{I}_u^- : f_{u,i^-} \geq \eta_{\alpha+d}\} = \{i^-_{(1)}, \ldots, i^-_{(K)}\}. \tag{47}$$

Therefore, as $B \to \infty$, constrained beam search selects negatives *exactly* from the top-d quantile set $\mathcal{Q}_u^-(\alpha+d)$ (equivalently, those with $f_{u,i^-} \geq \eta_{\alpha+d}$). In particular, if one defines the induced negative sampler $\mathbb{P}_{\text{neg}}(\cdot \mid u)$ by uniformly drawing from the returned Top-K beam outputs, then $\mathbb{P}_{\text{neg}}(\cdot \mid u)$ is precisely the uniform distribution over $\mathcal{Q}_u^-(\alpha+d)$ in the $B \to \infty$ limit. This establishes the claimed equivalence. □

G Proof of Proposition 3.3

Proof. By Lemma 3.1, GRPO induces the pairwise ranking push

$$\Pr_{i^+\sim\mathcal{I}_u^+,\, i^-\sim\mathbb{P}_{\text{neg}}(\cdot|u)} \big[f_{u,i^+} > f_{u,i^-}\big], \tag{48}$$

where $\mathbb{P}_{\text{neg}}(\cdot \mid u)$ is the item-level negative sampler determined by the decoding strategy.

When negatives are generated by constrained beam search, Lemma 3.2 implies that (in the hard-negative limit) the induced sampler equals the distribution over the top-d quantile negative set,

$$\mathbb{P}_{\text{neg}}(\cdot \mid u) \;=\; \mathbb{P}_{\text{hard}}(\cdot \mid u), \qquad \operatorname{supp}\big(\mathbb{P}_{\text{hard}}(\cdot \mid u)\big) \subseteq \mathcal{Q}_u^-(\alpha + d), \tag{49}$$

with $\mathcal{Q}_u^-(\alpha+d) = \{i^- \in \mathcal{I}_u^- : f_{u,i^-} \geq \eta_{\alpha+d}\}$ defined in Eq. (10). Substituting this into the induced pairwise objective yields

$$\Pr_{i^+\sim\mathcal{I}_u^+,\, i^-\sim\mathbb{P}_{\text{hard}}(\cdot|u)} \big[f_{u,i^+} > f_{u,i^-}\big]. \tag{50}$$

By definition of $\eta_{\alpha+d}$ in Eq. (6), sampling $j \sim \mathbb{P}_{\text{hard}}(\cdot \mid u)$ is equivalent to sampling negatives conditional on belonging to the upper-tail event $\{f_{u,i^-} \geq \eta_{\alpha+d}\}$, i.e., restricting comparisons to $\mathcal{Q}_u^-(\alpha+d)$. Therefore the above pairwise probability is exactly the one-way partial AUC at level d, namely $\mathrm{OPAUC}(\alpha+d)$. This proves the claim. □

G.1 Consistency Conditions for Connecting Sequence-Level Optimization to Item-Level Ranking

This section clarifies the consistency conditions required to connect sequence-level policy optimization with item-level pairwise ranking. In our setting, the transition from sequence-level ranking to item-level ranking relies on two conditions.

First, there exists a one-to-one mapping between each valid decoded token sequence Y and its corresponding item i, denoted by $i = \phi(Y)$. Under this condition, sampling over valid sequences can be equivalently viewed as sampling over items.

Second, the induced item score $f_{u,i}$ should be order-consistent with the sequence-level score $s_\theta(Y, H_u)$. Specifically, for a valid sequence Y corresponding to item $i = \phi(Y)$, we define

$$f_{u,i} = \prod_{j=1}^{\|Y\|} \pi_\theta(y_j \mid y_{<j}, H_u), \qquad i = \phi(Y). \tag{51}$$

The sequence-level ranking transition is exact when the ordering induced by $s_\theta(Y, H_u)$ is consistent with that induced by $f_{u,i}$.

To make this condition explicit, we compare the sequence-level score functions induced by different policy-gradient-based objectives. Let

$$\rho_{j,u} = \frac{\pi_\theta(y_j \mid y_{<j}, H_u)}{\pi_{\text{old}}(y_j \mid y_{<j}, H_u)}. \tag{52}$$

For vanilla policy gradient, the objective can be written as

$$\mathbb{E}_{H_u,\, Y\sim\pi_{\text{old}}(\cdot|H_u)} \left[\frac{1}{\|Y\|} \sum_{j=1}^{\|Y\|} \log \pi_\theta(y_j \mid y_{<j}, H_u) \hat{A}_j \right], \tag{53}$$

which induces the sequence-level score

$$s_\theta(Y, H_u) = \frac{1}{\|Y\|} \sum_{j=1}^{\|Y\|} \log \pi_\theta(y_j \mid y_{<j}, H_u). \tag{54}$$

Since

$$s_\theta(Y, H_u) = \frac{1}{\|Y\|} \log \prod_{j=1}^{\|Y\|} \pi_\theta(y_j \mid y_{<j}, H_u) = \frac{1}{\|Y\|} \log f_{u,i}, \tag{55}$$

the sequence-level score is exactly a monotone transformation of the induced item score. Therefore, for vanilla policy gradient, the transition from sequence-level pairwise ranking to item-level pairwise ranking is theoretically exact.

For GRPO, the clipped objective is

$$\mathbb{E}_{H_u,\, Y\sim\pi_{\mathrm{old}}(\cdot|H_u)}\left[\frac{1}{\|Y\|}\sum_{j=1}^{\|Y\|}\min\left(\rho_{j,u}\hat{A}_j,\ \mathrm{clip}(\rho_{j,u},1-\epsilon,1+\epsilon)\hat{A}_j\right)\right]. \tag{56}$$

This yields different effective sequence-level scores for positive and negative samples:

$$s_\theta^{\pm}(Y^{\pm},H_u)\triangleq\frac{1}{\|Y\|}\sum_{j=1}^{\|Y\|}\begin{cases}\min(\rho_{j,u},1+\epsilon), & (+),\\ \max(\rho_{j,u},1-\epsilon), & (-).\end{cases} \tag{57}$$

Unlike vanilla policy gradient, these clipped token-wise scores are not globally equivalent to a monotone transformation of $f_{u,i}$. The clipping operation can saturate individual token ratios and therefore may break global monotonicity between the sequence-level score and the induced item-level probability. Consequently, the connection between GRPO and item-level AUC should be understood as an approximate induced relationship rather than an exact equivalence.

Under the one-to-one mapping condition and the order-consistency approximation, the sequence-level pairwise push induced by GRPO can be interpreted at the item level as encouraging

$$\Pr_{i^+\sim\mathcal{I}_u^+,\, i^-\sim\mathbb{P}_{\mathrm{neg}}(\cdot|u)}\left[f_{u,i^+}>f_{u,i^-}\right]. \tag{58}$$

This formulation highlights the role of the decoding strategy. Different decoding strategies induce different negative-item distributions $\mathbb{P}_{\mathrm{neg}}(\cdot\mid u)$. In particular, constrained beam search concentrates the negative distribution on high-scoring, upper-tail negatives, thereby changing the effective pairwise ranking objective from a standard AUC-like objective toward a partial-AUC-like objective over hard negatives. This explains why the choice of negative generation strategy directly affects the implicit ranking behavior of GRPO.

H Proof of Theorem 3.4

Proof. Let $n\coloneqq n^+$ and let i be the number of positives in the Top-K prefix, so $\mathrm{Recall@K}=i/n$. Set

$$\alpha=\frac{K-n}{n^-},\qquad d=\frac{n}{n^-}, \tag{59}$$

then (up to the ceiling operators) $\mathcal{W}^-(\alpha,d)$ contains exactly the negatives with ranks

$$K-n<r\le K, \tag{60}$$

hence $|\mathcal{W}^-(\alpha,d)|=n$. Moreover, the top $K-n$ negatives $\{i^-_{(r)}\}_{r\le K-n}$ *always* appear before all window negatives in score order, so the Top-K event is fully determined by the relative order of the $2n$ items in $\mathcal{I}^+\cup\mathcal{W}^-(\alpha,d)$: among these $2n$ items, exactly i positives must fall in the top n positions.

Under this condition, $\mathrm{WPAUC}(\alpha,d)$ is simply the fraction of correctly ordered $(+,-)$ pairs within this $+/-$ sequence of length $2n$ (where $-$ denotes a window negative). We now bound it by extremal arrangements.

Maximum. The maximum $\mathrm{WPAUC}(\alpha,d)$ given i is attained when the first half contains i leading positives:

$$\underbrace{+\cdots+}_{i}\ \underbrace{-\cdots-}_{n-i}\ \Big|\ \underbrace{+\cdots+}_{n-i}\ \underbrace{-\cdots-}_{i}.$$

Then the number of concordant pairs is $in+(n-i)i=i(2n-i)$, hence

$$\mathrm{WPAUC}(\alpha,d)\ \le\ \frac{i(2n-i)}{n^2}. \tag{61}$$

Minimum. The minimum is attained when the first half contains i trailing positives:

$$\underbrace{-\cdots-}_{n-i}\ \underbrace{+\cdots+}_{i}\ \Big|\ \underbrace{-\cdots-}_{i}\ \underbrace{+\cdots+}_{n-i},$$

yielding exactly i^2 concordant pairs, hence

$$\mathrm{WPAUC}(\alpha, d)\ \geq\ \frac{i^2}{n^2}. \tag{62}$$

Let $w \coloneqq \mathrm{WPAUC}(\alpha, d)$. Since i is integer, inverting the above gives

$$\Big\lceil n\big(1-\sqrt{1-w}\big)\Big\rceil\ \leq\ i\ \leq\ \Big\lfloor n\sqrt{w}\Big\rfloor, \tag{63}$$

and dividing by n yields (13).

Tightness vs. $\mathrm{OPAUC}(\alpha{+}d)$**.** Note that $\alpha{+}d = \frac{K}{n^-}$, hence $\mathrm{OPAUC}(\alpha{+}d)$ uniformly samples negatives from the top-K negative prefix $\{i^-_{(r)} : 1 \leq r \leq K\}$. With $\alpha = (K-n^+)/n^-, d = n^+/n^-$, split this prefix into two disjoint blocks

$$\mathcal{H}^- \coloneqq \{i^-_{(r)} : 1 \leq r \leq K-n\}, \qquad \mathcal{W}^- \coloneqq \{i^-_{(r)} : K-n < r \leq K\} = \mathcal{W}^-(\alpha, d),$$

where $|\mathcal{H}^-| = K-n$ and $|\mathcal{W}^-| = n$. Define

$$A \coloneqq \Pr_{i^+\sim\mathcal{I}^+,\, j\sim\mathcal{H}^-}\big[f(i^+) > f(j)\big], \qquad B \coloneqq \Pr_{i^+\sim\mathcal{I}^+,\, j\sim\mathcal{W}^-}\big[f(i^+) > f(j)\big] = \mathrm{WPAUC}(\alpha, d).$$

By uniform mixing over $\mathcal{H}^- \cup \mathcal{W}^-$, we have the exact decomposition

$$\mathrm{OPAUC}(\alpha{+}d)\ =\ \frac{K-n}{K}\,A\ +\ \frac{n}{K}\,B. \tag{64}$$

Moreover, since every hard negative has score no smaller than every window negative ($f(i^-_{(r)}) \geq f(i^-_{(s)})$ for $r \leq K-n < s \leq K$), for any fixed $i^+, \mathbb{1}\{f(i^+) > j\} = 1$ for $j \in \mathcal{H}^- \Rightarrow \mathbb{1}\{f(i^+) > j'\} = 1$ for $j' \in \mathcal{W}^-$, and averaging yields the monotone constraint

$$0 \leq A \leq B \leq 1. \tag{65}$$

Let $o \coloneqq \mathrm{OPAUC}(\alpha{+}d) \in (0,1)$. From (64)–(65),

$$o \leq \frac{K-n}{K}B + \frac{n}{K}B = B \quad\Rightarrow\quad B \geq o, \qquad o \geq \frac{n}{K}B \quad\Rightarrow\quad B \leq \min\Big\{1, \frac{K}{n}o\Big\}.$$

Hence $\mathrm{OPAUC}(\alpha{+}d)$ only localizes the window statistic B to a *non-degenerate* interval

$$B \in \Big[o,\ \min\big\{1, \tfrac{K}{n}o\big\}\Big], \tag{66}$$

whose length is strictly positive for $o \in (0,1)$ since $K > n$.

Now denote the WPAUC-induced Recall bounds in (13) by

$$\underline{R}(B)\ \leq\ \mathrm{Recall@K}\ \leq\ \overline{R}(B),$$

where $\underline{R}(\cdot)$ and $\overline{R}(\cdot)$ are monotone nondecreasing in B (as proved above). If we only observe $o = \mathrm{OPAUC}(\alpha{+}d)$, then B may take any value in (66), so the set of Recall@K consistent with o must contain

$$\bigcup_{B\in[\,o,\ \min\{1,\frac{K}{n}o\}\,]} [\underline{R}(B), \overline{R}(B)]\ =\ \big[\underline{R}(o),\ \overline{R}(\min\{1, \tfrac{K}{n}o\})\big],$$

which is a strictly larger interval than $[\underline{R}(B), \overline{R}(B)]$ for any particular B (including the true $B = \mathrm{WPAUC}(\alpha, d)$), because (66) has positive length and $\underline{R}, \overline{R}$ are non-constant on $(0,1)$. Therefore, for the same K, the interval induced by (13) (conditioning on B) is strictly tighter than any interval obtainable from $\mathrm{OPAUC}(\alpha{+}d)$ alone (which conditions only on o and must account for the uncertainty in B).

□

Algorithm 2 TAWin: Threshold-Adjusted Soft Window Reweighting

Input: flattened token-level advantages $\boldsymbol{a} \in \mathbb{R}^N$, group size G, Top-K mass K, temperature $\tau > 0$, anchor $\tilde{\sigma}_\star \in [0, 1]$.
Output: reweighted advantages $\boldsymbol{a}' \in \mathbb{R}^N$.
Set $\boldsymbol{a}' \leftarrow \boldsymbol{a}$ and $M \leftarrow N/G$ {M groups}
for $g = 1$ **to** M **do**
  $\mathcal{I} \leftarrow \{(g-1)G + 1, \ldots, gG\}$; $\boldsymbol{a}_g \leftarrow \boldsymbol{a}'[\mathcal{I}]$
  $\mathcal{N} \leftarrow \{i \in \mathcal{I} : a_{g,i} < 0\}$; $\mathcal{P} \leftarrow \{i \in \mathcal{I} : a_{g,i} \geq 0\}$
  if $|\mathcal{N}| = 0$ **then**
    continue
  end if
  $K \leftarrow |\mathcal{N}|$
  Rank negatives: sort $\mathcal{N}$ by model score f in descending order and assign ranks $\sigma \in \{1, \ldots, K\}$ {higher score $\Rightarrow$ smaller rank}
  Normalize ranks: $\tilde{\sigma}(\sigma) \leftarrow \frac{\sigma - 1}{K - 1}$; if $K = 1$, define $\tilde{\sigma}(\sigma) \leftarrow 0$
  TAWin logits: for each negative with rank r, set

$$x(\sigma) \leftarrow -\big|\tilde{\sigma}(\sigma) - \tilde{\sigma}(\sigma_\star)\big|$$

  Sort logits: let $x_{(1)} \geq \cdots \geq x_{(K)}$ denote logits in non-increasing order
  Compute threshold λ: find an index $m \in \{0, \ldots, K-1\}$ such that $x_{(m)} \geq \lambda \geq x_{(m+1)}$ (with $x_{(0)} = +\infty$), and set

$$\lambda \leftarrow \tau \log\Big(\sum_{i=m+1}^{K} \exp\big(\tfrac{x_{(i)}}{\tau}\big) \Big) - \tau \log(K - m)$$

  ThreTopK weights (sorted order): for $i = 1, \ldots, K$,

$$w_{(i)} \leftarrow \min\Big\{1, \exp\big(\tfrac{x_{(i)} - \lambda}{\tau}\big)\Big\}$$

  Restore original order: map $\{w_{(i)}\}$ back to indices $i \in \mathcal{N}$ to obtain $\{w_i\}_{i \in \mathcal{N}}$
  Mass normalization: for all $i \in \mathcal{N}$, set

$$\bar{w}_i \leftarrow \frac{w_i}{\sum_{j \in \mathcal{N}} w_j} \cdot |\mathcal{N}|$$

  Reweight negative advantages: $a'_{g,i} \leftarrow a'_{g,i} \cdot \bar{w}_i \quad \forall i \in \mathcal{N}$
end for
return $\boldsymbol{a}'$

H.1 Proof of Lemma 3.5

Proof. Let the unique positive item be i^+. By definition, $\mathrm{Recall@K} = \mathbb{I}\{\mathrm{rank}(i^+) \leq K\}$.

The windowed negative set $\mathcal{W}_u^-(\alpha, d)$ in (11) contains exactly one negative item: the $(K-1)$-th highest-scored negative, denoted $j_{(K-1)}$. Therefore, by (12),

$$\mathrm{WPAUC}_u(\alpha, d) = \mathbb{I}\{f_{u,i^+} > f_{u,j_{(K-1)}}\}.$$

The inequality $f_{u,i^+} > f_{u,j_{(K-1)}}$ holds iff the positive item is ranked above the $(K-1)$-th negative, which is equivalent to i^+ appearing within the overall top-K positions. Hence $\mathrm{WPAUC}_u(\alpha, d) = \mathbb{I}\{\mathrm{rank}(i^+) \leq K\} = \mathrm{Recall@K}$. □

I TAWin Algorithm

The detailed algorithm of TAWin is provided in 2.

Datasets	Toys	Industrial	Office	Yelp
Items	11,252	3,685	3,459	8,785
Train	112,754	36,259	38,924	77,097
Valid	14,095	4,532	4,866	9,637
Test	14,095	4,533	4,866	9,638

Table 5: Statistics of datasets.

J Experimental Settings

J.1 Dataset

We construct our datasets from the Amazon Review corpus following the preprocessing pipeline in Bao et al. [2], Tan et al. [40]. First, we restrict interactions to a fixed temporal window defined per category: from October 2016 to November 2018 for Toys and Games and Office Products, and from October 1996 to November 2018 for Industrial and Scientific. For Yelp, we select the interactions from the year of 2021. We then apply iterative K-core filtering with $K = 5$, removing users and items with fewer than five interactions until convergence. To ensure high-quality textual representations, we further discard items with missing or noisy metadata and retain only those with valid and concise titles.

For each remaining user, interactions are sorted chronologically and transformed into supervised samples using a sliding window, where the historical context is truncated to at most ten preceding items and the subsequent item is treated as the prediction target. This procedure yields multiple training instances per user while preserving temporal order. Finally, all samples are globally sorted by timestamp and split chronologically into training, validation, and test sets using an 8:1:1 ratio. Dataset statistics for the resulting training splits are reported in Table 5.

J.2 Evaluation Metrics.

To assess Top-K recommendation performance, we adopt **Recall (R@K)** and **Normalized Discounted Cumulative Gain (N@K)**. For each user u, we apply the constrained beam search generation strategy $\mathcal{G}$ to obtain a ranked list of K recommended items:

$$\mathcal{R}_u^K \triangleq \big\{\phi(Y_1), \ldots, \phi(Y_K)\big\}, \tag{67}$$

where $\{Y_1, \ldots, Y_K\}$ are the Top-K completed sequences returned by $\mathcal{G}(\pi_\theta, H_u, \mathcal{I})$, ordered by model likelihood.

Recall (R@K). R@K measures whether the ground-truth target item i_t appears in the Top-K recommendation list:

$$\mathrm{R@}K \triangleq \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbb{I}\big[\, i_t \in \mathcal{R}_u^K \,\big], \tag{68}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function.

Normalized Discounted Cumulative Gain (N@K). N@K further accounts for the rank position of the target item within the Top-K list. Let $\mathrm{rank}_u(i_t)$ denote the rank of i_t in $\mathcal{R}_u^K$ (smaller is better). Then

$$\mathrm{N@}K \triangleq \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{\mathbb{I}\big[\, i_t \in \mathcal{R}_u^K \,\big]}{\log_2\big(\mathrm{rank}_u(i_t) + 1\big)}. \tag{69}$$

J.3 Baselines

We compare TAWin with representative baselines from three categories: traditional sequential recommendation models, generative recommendation models, and LLM-based recommendation models. The baselines are described as follows:

- **GRU4Rec** [13] models session-based recommendation by applying recurrent neural networks to entire interaction sequences, enabling temporal dependency modeling with a ranking-oriented loss.
- **Caser** [41] embeds recent interaction sequences into a two-dimensional representation and applies horizontal and vertical convolutional filters to capture both general preferences and local sequential patterns.
- **SASRec** [15] employs a self-attention mechanism to adaptively select relevant items from a user's interaction history, balancing long-term dependency modeling and efficient sequential prediction.
- **TIGER** [31] formulates recommendation as a generative retrieval task by predicting semantically structured item identifiers using a Transformer-based sequence-to-sequence model.
- **LC-Rec** [50] bridges language and collaborative semantics by learning quantized item indices and aligning LLMs to recommendation tasks through specialized fine-tuning objectives.
- **MiniOneRec** [16] presents an open-source generative recommendation framework that constructs Semantic IDs and applies supervised and reinforcement learning to study scaling behaviors of LLM-based recommenders.
- **BigRec** [3] proposes a bi-step grounding paradigm that first aligns LLM outputs to recommendation tokens and then maps them to valid items to achieve full-corpus ranking capability.
- **D**^{3} [2] introduces a debiasing and diversifying decoding strategy that removes length normalization effects and incorporates auxiliary models to improve accuracy and recommendation diversity.
- **S-DPO** [7] extends Direct Preference Optimization to recommendation by incorporating multiple negatives and optimizing a ranking-aware objective derived from partial Plackett–Luce models.
- **ReRe** [40] optimizes LLM-based recommenders via reinforcement learning with verifiable rewards, combining constrained generation and auxiliary ranking supervision to improve hard-negative modeling.

J.4 Implement Details

Traditional recommendation models are trained with binary cross-entropy loss and the Adam optimizer. The learning rate is tuned over $\{10^{-2}, 10^{-3}, 10^{-4}\}$ and the weight decay is selected from $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$; the batch size is fixed to 1024. For TIGER, we adopt T5 [30] as the backbone.

For LLM-based recommenders, we use Qwen2.5-0.5B [29] as the base model to reduce computational overhead, and optimize all models with AdamW [23]. During fine-tuning, SFT and preference-alignment data are processed with batch size 128, while reinforcement learning uses batch size 512. We apply a learning rate of 3×10^{-4} for SFT and 1×10^{-5} for S-DPO, SPRec, and ReRe, all with a cosine learning-rate scheduler. SFT runs for up to 10 epochs with early stopping (patience 1). DPO-based models (S-DPO, SPRec) are trained for 1 epoch, while ReRe is trained for 2 epochs.

For D^3, the interpolation coefficient α is searched over $\{0.8, 0.9, 1.0\}$. For DPO-based methods, we set the KL coefficient to $\beta = 0.1$, and S-DPO samples 3 negative items per instance. For ReRe, we set $\beta = 10^{-3}$ and generate 16 candidate items per prompt. When implementing GRPO, we estimate the KL divergence via a second-order approximation.

J.5 Practical Window Selection For TAWin.

For a new dataset or recommendation target, we suggest starting with $\sigma_\star = 1$ and $\hat{K} = G/4$ ($\hat{K} = 4$ for $G = 16$ rollouts). Increase the anchor $\sigma_\star$ when larger target cutoffs matter, and decrease it for smaller cutoffs. A larger $\hat{K}$ emphasizes a broader range of cutoffs, whereas a smaller $\hat{K}$ concentrates on a narrower target range. These target-dependent directions reduce the validation search, and the same settings can serve as a starting point across datasets with matching evaluation targets.

Table 6: Five-seed results on all four datasets (mean $\pm$ standard deviation). Bold marks the highest mean; † indicates a reported significant improvement over the strongest baseline at $p < 0.01$ under a paired t-test.

Dataset	Method	R@1	R@3	N@3
Toys	ReRe	0.0419 ± 0.0005	0.0709 ± 0.0009	0.0588 ± 0.0006
	ReRe*	0.0431 ± 0.0007	0.0723 ± 0.0005	0.0601 ± 0.0004
	TAWin	$\mathbf{0.0465}^\dagger \pm 0.0007$	$\mathbf{0.0734}^\dagger \pm 0.0004$	$\mathbf{0.0621}^\dagger \pm 0.0005$
Industrial	ReRe	0.0812 ± 0.0007	0.1221 ± 0.0011	0.1049 ± 0.0009
	ReRe*	0.0829 ± 0.0005	0.1209 ± 0.0005	0.1050 ± 0.0005
	TAWin	$\mathbf{0.0891}^\dagger \pm 0.0005$	$\mathbf{0.1227} \pm 0.0009$	$\mathbf{0.1087}^\dagger \pm 0.0006$
Office	ReRe	0.0834 ± 0.0007	0.1307 ± 0.0003	0.1110 ± 0.0004
	ReRe*	0.0892 ± 0.0006	0.1325 ± 0.0005	0.1153 ± 0.0005
	TAWin	$\mathbf{0.0956}^\dagger \pm 0.0004$	$\mathbf{0.1346}^\dagger \pm 0.0006$	$\mathbf{0.1187}^\dagger \pm 0.0004$
Yelp	ReRe	0.0209 ± 0.0005	0.0353 ± 0.0006	0.0293 ± 0.0003
	ReRe*	0.0211 ± 0.0006	0.0357 ± 0.0007	0.0296 ± 0.0005
	TAWin	$\mathbf{0.0228}^\dagger \pm 0.0006$	$\mathbf{0.0370}^\dagger \pm 0.0005$	$\mathbf{0.0311}^\dagger \pm 0.0005$

Table 7: Performance comparison between ReRe with ranking reward (ReRe*) and TAWin. The best performance is highlighted in boldface.

Method	Toys			Industrial			Office			Yelp		
	R@1	R@3	N@3	R@1	R@3	N@3	R@1	R@3	N@3	R@1	R@3	N@3
ReRe*	0.0442	0.0740	0.0615	0.0882	0.1224	0.1091	0.0931	0.1333	0.1170	0.0206	0.0358	0.0295
TAWin	**0.0471**	**0.0761**	**0.0639**	**0.0904**	**0.1237**	**0.1099**	**0.0961**	**0.1341**	**0.1187**	**0.0227**	**0.0370**	**0.0301**

They provide practical tuning guidance rather than an automatic mapping from target K to window parameters.

K Significance Analysis

We compare ReRe, its ranking-reward variant ReRe*, and TAWin on all four datasets using five random seeds: 42, 43, 44, 45, and 46. These supplementary runs use 8 NVIDIA H100 GPUs rather than the H800 GPUs used for the original experiments, with all other reported hyperparameters unchanged. Table 6 reports the mean and standard deviation across seeds, together with the reported paired t-tests against the strongest baseline for each dataset–metric setting.

TAWin achieves the highest mean in all 12 dataset–metric settings, with significant improvements in 11 settings. The exception is Industrial R@3, where ReRe is the strongest baseline (0.1221 versus 0.1209 for ReRe*) and TAWin reaches 0.1227 without a significance marker. Averaging relative improvements over the strongest baseline for each metric and dataset yields 7.65%, 1.81%, and 3.72% for R@1, R@3, and N@3, respectively. These relative improvements quantify the magnitude of the observed gains, while the five-seed standard deviations characterize run-to-run variation. The H100 reruns are reported separately from the original H800 point estimates.

L Reweighting Ablation

While ReRe* offers an intuitive heuristic that favors hard negatives identified by beam search, its weighting rule is not derived from an explicit Top-K alignment objective. In contrast, TAWin is directly motivated by our theoretical insight that Top-K optimization can be better approached via windowed partial-AUC shaping, and it implements a differentiable, temperature-controlled reweighting that targets the desired low-FPR region. As reported in Table 7, TAWin consistently outperforms ReRe* across four datasets (Toys, Industrial, Office, and Yelp) in terms of both Recall and NDCG at small cutoffs, demonstrating that TAWin yields more effective preference optimization than the rank-based heuristic used in ReRe*.

Table 8: ReRe and TAWin with larger backbones on three datasets. Bold marks the better result for each backbone, dataset, and metric.

Backbone	Method	Toys			Industrial			Office		
		R@1	R@3	N@3	R@1	R@3	N@3	R@1	R@3	N@3
Qwen2.5-3B	ReRe	0.0452	0.0729	0.0612	0.0825	0.1244	0.1066	0.0881	0.1326	0.1143
	TAWin	**0.0481**	**0.0764**	**0.0644**	**0.0909**	**0.1257**	**0.1099**	**0.0978**	**0.1334**	**0.1190**
Qwen2.5-7B	ReRe	0.0466	0.0742	0.0627	0.0878	0.1253	0.1099	0.0927	0.1326	0.1160
	TAWin	**0.0478**	**0.0749**	**0.0635**	**0.0922**	**0.1264**	**0.1107**	**0.1005**	**0.1344**	**0.1203**
Llama-3.2-1B	ReRe	0.0452	0.0734	0.0608	0.0849	0.1230	0.1070	0.0904	0.1300	0.1136
	TAWin	**0.0483**	**0.0766**	**0.0644**	**0.0941**	**0.1261**	**0.1115**	**0.0972**	**0.1313**	**0.1172**

Table 9: End-to-end training time on 8 NVIDIA H100 GPUs, reported as mean $\pm$ standard deviation over five runs.

Dataset	ReRe (min)	TAWin (min)	Relative difference
Toys	320.6 ± 1.8	344.0 ± 3.7	$+7.30\%$
Industrial	166.0 ± 1.6	172.4 ± 2.9	$+3.86\%$
Office	189.4 ± 1.5	187.8 ± 2.5	-0.84%
Yelp	203.6 ± 1.9	204.8 ± 6.3	$+0.59\%$

M Generalization to Larger Backbones

To assess whether the gains extend beyond the default 0.5B backbone, we compare ReRe and TAWin using Qwen2.5-3B, Qwen2.5-7B, and Llama-3.2-1B on Toys, Industrial, and Office. Table 8 reports the existing backbone-ablation results. TAWin outperforms ReRe in every reported dataset–metric setting across all three backbones, showing that the gains persist with larger models and across model families.

These results support the transferability of windowed objective shaping across the tested model scales. The magnitude of the gain varies across backbones, and increasing model size does not uniformly improve every metric. This comparison evaluates final recommendation performance; it does not include dedicated measurements of training instability, so we do not infer a stability guarantee from the accuracy results alone.

N Computational Efficiency

TAWin and ReRe share the same LLM backbone, GRPO training pipeline, generation process, and forward/backward passes. TAWin adds only the computation and application of windowed weights to a small candidate advantage vector ($G = 16$), without additional model forward passes or activation storage. The additional arithmetic and vector-storage costs are small relative to those of the LLM.

To assess practical efficiency, we measure end-to-end training time over five runs on 8 NVIDIA H100 GPUs. Table 9 reports the mean and standard deviation on all four datasets. The relative runtime difference is calculated as $100(t_{\mathrm{TAWin}} - t_{\mathrm{ReRe}})/t_{\mathrm{ReRe}}$. These measurements belong to the supplementary H100 experiments and are distinct from the original H800 experiments.

The observed runtime differences range from -0.84% to $+7.30\%$, indicating modest end-to-end overhead in the reported settings despite the small arithmetic cost of reweighting. The slightly shorter Office runtime is within the reported run-to-run variation and does not establish a systematic speedup. Overall, these measurements support the practical efficiency of incorporating TAWin into the existing RL pipeline.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect this paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Please refer to Appendix A.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: The complete proof of all theorems and lemmas can be found in the Appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide all necessary materials to reproduce the main results in Appendix J.

   Guidelines:

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

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

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

Answer:[Yes]

Justification: The code and data are available at https://anonymous.4open.science/r/TAWin-E6CE/.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide all necessary experimental settings to reproduce the main results in Appendix J.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Due to computational resource constraints, we did not compute significance bars for all results. Instead, we calculated them for a representative subset of results and report the corresponding findings in Appendix K.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: All experiments are conducted on 8 NVIDIA H800 GPUs. The detailed information can be found in Section 4.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research in this paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Please refer to Appendix B.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We correctly cite the original paper that produced the code package or dataset.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.