
On Efficiency-Effectiveness Trade-off of Diffusion-based Recommenders

Wenyu Mao¹, Jiancan Wu^{1,2*}, Guoqing Hu¹, Zhengyi Yang¹, Wei Ji³, Xiang Wang^{1*},

¹ University of Science and Technology of China

² Institute of Dataspace, Hefei Comprehensive National Science Center

³ Nanjing University

Abstract

Diffusion models have emerged as a powerful paradigm for generative sequential recommendation, which typically generate next items to recommend guided by user interaction histories with a multi-step denoising process. However, the multi-step process relies on discrete approximations, introducing discretization error that creates a trade-off between computational efficiency and recommendation effectiveness. To address this trade-off, we propose TA-Rec, a two-stage framework that achieves one-step generation by smoothing the denoising function during pretraining while alleviating trajectory deviation by aligning with user preferences during fine-tuning. Specifically, to improve the efficiency without sacrificing the recommendation performance, TA-Rec pretrains the denoising model with Temporal Consistency Regularization (TCR), enforcing the consistency between the denoising results across adjacent steps. Thus, we can smooth the denoising function to map the noise as oracle items in one step with bounded error. To further enhance effectiveness, TA-Rec introduces Adaptive Preference Alignment (APA) that aligns the denoising process with user preference adaptively based on preference pair similarity and timesteps. Extensive experiments prove that TA-Rec’s two-stage objective effectively mitigates the discretization errors-induced trade-off, enhancing both efficiency and effectiveness of diffusion-based recommenders. Our code is available at <https://github.com/maowenyu-11/TA-Rec>.

1 Introduction

Diffusion models [1–3] have recently shown strong potential in generative sequential recommendation [4–6, 4, 7], owing to their remarkable ability to model complex distributions of user behaviors and generate oracle items that best match user preferences. At the core of diffusion-based sequential recommendation [4, 5, 7] is framing a theoretical continuous process [2, 3] which gradually reconstructs the oracle items from random noise. Typically, such theoretical continuous process (*e.g.*, Stochastic Differential Equations (SDE) [2]) is realized by multi-step discrete approximations [8–11] (*e.g.*, Denoising Diffusion Probabilistic Models (DDPM) [1]) for practical implementation: they add noise to the ground-truth next items in the forward process and then denoise it step by step in the reverse process with denoising models, guided by the interaction history [12].

While achieving success, this discrete approximation inherently introduces discretization errors [13] between the practical reverse trajectory (*i.e.*, the dashed line of discrete DDPM in Figure 1a) and the theoretical trajectory (*i.e.*, the solid line of continuous SDE in Figure 1a). Such discretization error is primarily caused by high-order truncation in numerical solvers and accumulates across the multi-step reverse process. When approximating SDE [2] with finite steps T , discretization errors grow as T decreases [14, 15], creating an efficiency-effectiveness trade-off: increasing the number of steps T

*Corresponding author: wujcan@gmail.com, xiangwang@ustc.edu.cn.

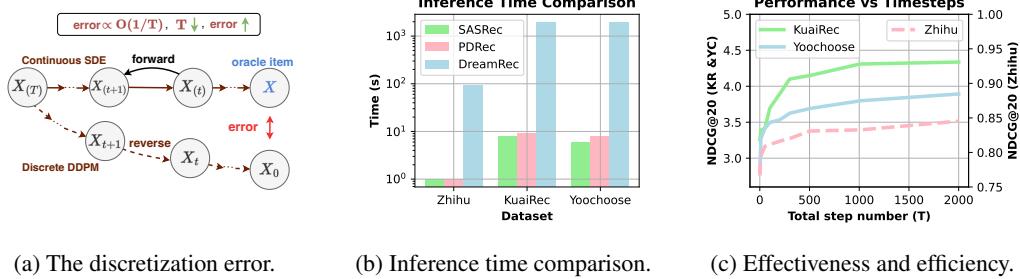


Figure 1: Overall motivation for our work, including the existence of discretization error, high inference costs for diffusion-based recommenders, and the effectiveness and efficiency trade-off.

(typically 1,000 steps) for diffusion-based recommenders improves alignment between generated items and user preferences but incurs substantial inference overhead, while aggressively reducing total steps T accelerates generation at the cost of amplified discretization error and misalignment. As shown in Figure 1b, compared with traditional recommenders (e.g., SASRec [16] and PDRec [17]), the diffusion-based DreamRec [4] requires 1,000 steps to achieve comparable performance, significantly slowing inference. Conversely, reducing T significantly boosts acceleration yet leads to notable performance drops of DreamRec — evidenced by a significant NDCG drop from $T = 2,000$ to $T = 1$ on different datasets in Figure 1c.

To resolve the trade-off, we propose TA-Rec, a two-stage framework that realizes one-step generation by smoothing the denoising function in pretraining stage and reduces the trajectory deviation by optimizing denoising models with user preferences in fine-tuning stage. Technically, in the pretraining stage, TA-Rec proposes temporal consistency regularization (TCR) to enforce consistency between denoising representations at adjacent timesteps, smoothing the denoising function [18] to map noise to oracle items directly [15, 14] with bounded error. This could improve diffusion models’ efficiency in one-step generation without compromising the recommendation performance. In the fine-tuning stage, TA-Rec introduces adaptive preference alignment (APA) [7, 19] to mitigate the trajectory deviation and enhance recommenders’ effectiveness. Specifically, APA constructs preference pairs by selecting positive and negative items from the interaction space. Recognizing that the denoising trajectory from noise to oracle items inherently varies with both user preferences and noise magnitude, APA dynamically calibrates alignment strength according to both the similarity between preference pairs and the time steps of the noisy input, optimizing denoising models in a more refined manner. Notably, when the preference pair is hard to distinguish and the noisy input has a large noise degree, APA strategically reduces optimization strength to avoid overfitting to ambiguous preferences and random noise.

Theoretical analysis demonstrates that TA-Rec has bounded error despite accelerated generation and achieves more accurate alignment, thereby enabling reliable recommendations matching user preference. Empirical results across multiple real-world benchmarks (i.e., zhihu [20], KuaiRec [21], and YooChoose [22]) show that our approach achieves $100\times$ faster generation compared to leading diffusion-based recommenders (e.g., DreamRec [4]) while simultaneously improving recommendation performance by 10%, achieving both higher efficiency and higher effectiveness.

2 Related Work

Diffusion-based Recommenders [4, 6, 23–25] have emerged as a promising alternative to conventional sequential recommenders by modeling the complex user behavior distribution and generating next items step by step to match user preference. Existing methods have explored various applications of diffusion models in recommendation systems, including next-item generation through a denoising process guided by historical sequences [4, 5, 23], augmentation of sequential recommenders [26, 17, 27], and improvement of robustness [28] against noisy feedback [29, 30]. Our work addresses the issue of discretization error [13] in the reverse process for next-item generation paradigm, which fundamentally impacts the efficiency-effectiveness trade-off that currently limits diffusion-based recommenders’ practical deployment.

Accelerating Diffusion [14, 31–34] addresses the computational bottleneck inherent in the multi-step generation process of diffusion models. Recent advancements fall into two primary categories: training-free numerical solvers [14, 35, 36] and training-based model distillation [37–40]. The former

optimizes numerical solvers to serve as faster samplers during the reverse process, reducing inference steps to 20-50 without tuning the original denoising model [14, 41, 42]. For instance, DPM++ [43] leverages adaptive high-order solvers to approximate the continuous SDEs [2], achieving faster generation via optimized numerical integration. To achieve more aggressive acceleration (1-4 steps), the latter approach focuses on tuning the denoising model through knowledge distillation. Specifically, consistency models [15, 44] distill pre-trained diffusion models into single-step generators by enforcing self-consistency of the ODE trajectory. However, these methods may either rely on well-pretrained models or a deterministic ODE process, limiting their applicability to recommendation tasks due to the lack of unified recommendation benchmarks and the dynamic nature of user preferences. Instead, our work achieves one-step generation of diffusion-based recommenders by designing a temporal consistency loss for the stochastic DDPM process.

Aligning Diffusion Models [19, 45–47] aim to optimize diffusion models using human preference data, an area that remains relatively underdeveloped compared to preference optimization in large language models (LLMs). Current approaches often adapt existing LLM preference optimization algorithms to diffusion models. For example, Diffusion-DPO [19] takes the DPO loss [48] from LLMs and adapts it on diffusion models directly to improve image generation based on user preference. DSPO [47] aligns diffusion models with human preferences by distilling the score function of preferred image distributions into the model’s pretrained score functions, leveraging score matching. Furthermore, Diffusion-NPO [46] approaches preference alignment by training an additional model to specifically model negative preferences. To enable alignment at a more fine-grained level, our work designs an adaptive preference optimization algorithm that aligns the generation of diffusion-based recommenders according to both timesteps and pair similarity.

3 Preliminaries

3.1 Diffusion-based Sequential Recommendation

In sequential recommendation tasks, we represent a user’s historical interaction sequence chronologically as $\mathbf{x}^{1:N-1} = [\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{N-1}]$ in the embedding space, where each $\mathbf{x}^n \in \mathbb{R}^d$ denotes the embedding of the user’s n -th interacted item. The next item to be interacted with is denoted as \mathbf{x}^N (\mathbf{x} for simplification). The goal of diffusion-based sequential recommendation is to recover the oracle item [4] that best aligns with users’ preferences from noise, guided by user interaction history. Following [4, 23, 7], in the forward process, Gaussian noise is added to the next item \mathbf{x} : $q(\mathbf{x}_t|\mathbf{x}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}, (1 - \bar{\alpha}_t)\mathbf{I})$, where $[\alpha_1, \dots, \alpha_T]$ denotes the noise scale, $t \in [1, \dots, T]$ denotes the timestep. During the reverse process, diffusion models recover the next item \mathbf{x} from noise under the guidance \mathbf{g} through a multi-step denoising process. To optimize the denoising model, the training loss for diffusion-based recommenders can be formulated as [4, 7]:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{x}, \mathbf{g}, t} \left[\|f_{\theta}(\mathbf{x}_t, \mathbf{g}, t) - \mathbf{x}\|_2^2 \right], \quad (1)$$

where \mathbf{g} is the guidance signal extracted from users’ historical interaction sequences $\mathbf{x}^{1:N-1}$ with a Transformer following [4, 7], $f_{\theta}(\cdot)$ is a denoising model parameterized by MLP, **reconstructing** the target item \mathbf{x} as $\hat{\mathbf{x}}_0$ directly under the guidance \mathbf{g} , as shown in the green part of Figure 2. Commonly, classifier-free guidance paradigm [12] can be utilized by replacing the guidance \mathbf{g} with a dummy token Φ at probability ρ , allowing for the integration of conditional and unconditional training. During inference, the pure Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ serves as the input, then the denoising model $f_{\theta}(\cdot)$ denoises items as $\hat{\mathbf{x}}_0$ under the guidance \mathbf{g} . The iterative reverse process for generation is:

$$\mathbf{x}_{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} \hat{\mathbf{x}}_0 + \frac{\sqrt{\bar{\alpha}_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_t + \sqrt{\bar{\beta}_t} \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (2)$$

After generating the oracle item \mathbf{x}_0 step-by-step, we calculate the dot product between \mathbf{x}_0 and each candidate item in the corpus, then recommend K items having the highest similarity scores.

3.2 Direct Preference Optimization on Diffusion Models

Direct Preference Optimization (DPO) [49–51] is a reward-model-free method that aligns LLMs with human preferences via a supervised loss derived from pairwise comparison. To adapt DPO [48] to the diffusion model [19, 47, 46], the key idea is to increase denoising models’ likelihood of preferred

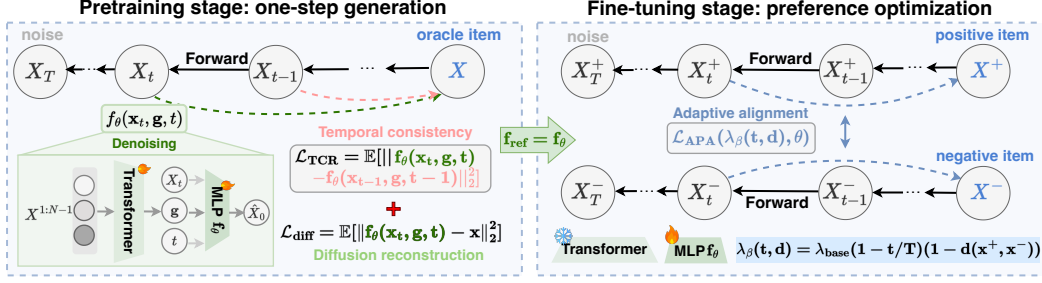


Figure 2: The overview of our proposed TA-Rec framework, which alleviates the efficiency and effectiveness trade-off of diffusion-based recommenders with two stages. In the pretraining stage, TA-Rec achieves one-step generation with temporal consistency regularization. In the fine-tuning stage, TA-Rec adopts an adaptive coefficient $\lambda_\beta(t, d)$ to align the generated item with users' preferences.

generation while decreasing that of dispreferred ones. We assume the pairwise preference data as $D = (\mathbf{x}^+, \mathbf{x}^-)$, where \mathbf{x}^+ is preferred over \mathbf{x}^- . Following [19], the DPO training loss on diffusion model can be formulated as:

$$\mathcal{L}_{\text{Diffusion-DPO}} = -\mathbb{E} \left[\log \sigma \left(\lambda_\beta \log \frac{p_\theta(\mathbf{x}_t^+ | \mathbf{x}_{t+1}^+, \mathbf{g})}{p_{\text{ref}}(\mathbf{x}_t^+ | \mathbf{x}_{t+1}^+, \mathbf{g})} - \lambda_\beta \log \frac{p_\theta(\mathbf{x}_t^- | \mathbf{x}_{t+1}^-, \mathbf{g})}{p_{\text{ref}}(\mathbf{x}_t^- | \mathbf{x}_{t+1}^-, \mathbf{g})} \right) \right], \quad (3)$$

where $\mathbf{x}_t^+ \sim q(\mathbf{x}_t^+ | \mathbf{x}_0^+)$, $\mathbf{x}_t^- \sim q(\mathbf{x}_t^- | \mathbf{x}_0^-)$, \mathbf{g} is the guidance of denoising model, p_{ref} represents the reference distribution, λ_β is a hyperparameter which controls preference optimization strength.

4 Method

In this section, we present our proposed TA-Rec framework, a two-stage approach addressing the discretization error-induced trade-off in diffusion-based recommenders. The overall framework is illustrated in Figure 2. We first define the discretization error of diffusion models in Section 4.1. We then detail the Temporal Consistency Regularization (TCR) during pretraining in Section 4.2. Finally, we describe the Adaptive Preference Alignment (APA) during fine-tuning in Section 4.3.

4.1 Definition of Discretization Error in Diffusion Models

The continuous reverse process of the diffusion model can be described by the SDE process [2] mathematically. Formally, we have: $d\mathbf{x} = [f(\mathbf{x}, s) - g(s)^2 \nabla_{\mathbf{x}} \log p_s(\mathbf{x})] ds + g(s) d\bar{w}$, where $f(\cdot)$ and $g(\cdot)$ are drift and diffusion coefficients, \bar{w} is reverse-time Brownian motion, and $p_s(\mathbf{x})$ is the marginal distribution at time s ($s \in [0, 1]$). In practice, the theoretical continuous SDE is approximated by discrete numerical solvers (e.g., Euler-Maruyama [52], a first-order solver) with T steps (step size $\Delta t = 1/T$): $\mathbf{x}_{t+1} = \mathbf{x}_t + [f - g^2 \nabla \log p_t] \Delta t + g \sqrt{\Delta t} \mathbf{z}$, $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$.

Definition 1 The deviation between the continuous process and its discrete approximation is the discretization error $\mathcal{E}_{\text{disc}}$: $\mathcal{E}_{\text{disc}}(t) = \mathbb{E}_{\mathbf{x} \sim p(t)} [\|\mathbf{x}_{(t)} - \mathbf{x}_t\|_2]$, where $\mathbf{x}_{(t)}$ is the theoretical SDE solution at time $s = t \cdot \Delta t$, \mathbf{x}_t is the discrete approximation at step t , $t \in [1, \dots, T]$.

Such discretization error primarily stems from local truncation error in solvers' numerical integration [53, 54] and is accumulated globally during the propagation across multiple steps in the reverse process [55, 56], where $\mathcal{E}_{\text{disc}}(0) \sim O(\Delta t)$. Thus, reducing reverse steps T (i.e., increasing Δt) will amplify $\mathcal{E}_{\text{disc}}$ and cause trajectory deviation, introducing the efficiency and effectiveness trade-off.

4.2 Pretraining Stage: Temporal Consistency Regularization (TCR)

In recommendation systems, the multi-step reverse process introduces high computational costs, failing to meet the low-latency demands of large-scale online deployments. To improve the efficiency of diffusion-based recommenders, we aim to realize one-step generation, which maps the noise to the oracle items directly. Specifically, TA-Rec proposes temporal consistency regularization (TCR) to smooth the denoising function of diffusion-based recommenders in the pretraining stage, as shown

in the left part of Figure 2. In addition to the reconstruction loss in Equation (1), TCR enforces consistency between consecutive timesteps to smooth the denoising function $f_\theta(\cdot)$ with Lipschitz continuity [18]. For adjacent noisy representations \mathbf{x}_t and \mathbf{x}_{t-1} (obtained via the forward process $q(\mathbf{x}_t|\mathbf{x}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}, (1 - \bar{\alpha}_t)\mathbf{I})$ as introduced in Section 3.1), the TCR loss can be defined as:

$$\mathcal{L}_{\text{TCR}} = \mathbb{E}_{\mathbf{x}_t, \mathbf{g}, t} [\|f_\theta(\mathbf{x}_t, \mathbf{g}, t) - f_\theta(\mathbf{x}_{t-1}, \mathbf{g}, t-1)\|_2^2], \quad (4)$$

where \mathbf{g} is the guidance signal extracted from the historical interaction sequence $\mathbf{x}^{1:N-1}$ with a Transformer, as introduced in Section 3.1. The total loss, which jointly optimizes the denoising model with reconstruction loss in Equation (1) and TCR loss in Equation (4), can be formulated as:

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{diff}} + \lambda_c \cdot \mathcal{L}_{\text{TCR}}, \quad (5)$$

where λ_c is the strength that balances reconstruction accuracy and denoising smoothness. Upon optimizing \mathcal{L}_{pre} , we can obtain the smooth denoising function $f_\theta(\mathbf{x}_t, \mathbf{g}, t)$ that consistently projects the noisy representation \mathbf{x}_t at any step on the trajectory to the target item \mathbf{x} . Thus, the pure noise $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ can be mapped by the smooth denoising function as oracle items \mathbf{x}_0 directly with one-step generation, eliminating the need for iterative reverse steps. Below, we justify that such accelerated generation has a bounded error.

Theorem 1 (Error Bound for One-Step Generation) *We assume that: (i) The smooth denoising function $f_\theta(\mathbf{x}_t, \mathbf{g}, t)$ satisfies Lipschitz continuity with constants $L > 0$: $\|f_\theta(\mathbf{x}_{s_1/\Delta t}, \mathbf{g}, s_1/\Delta t) - f_\theta(\mathbf{x}_{s_2/\Delta t}, \mathbf{g}, s_2/\Delta t)\|_2 \leq L|s_1 - s_2|$ for any time $s_1, s_2 \in [0, 1]$. (ii) For any step $t \in [1, \dots, T]$ and any time $s \in [t\Delta t, (t+1)\Delta t]$, there exists a constant C such that the following smooth condition holds $\|f_\theta(\mathbf{x}_{s/\Delta t}, \mathbf{g}, s/\Delta t) - f_\theta(\mathbf{x}_t, \mathbf{g}, t)\|_2 \leq C\|f_\theta(\mathbf{x}_{t+1}, \mathbf{g}, t+1) - f_\theta(\mathbf{x}_t, \mathbf{g}, t)\|_2$. Then, since we minimize L_{pre} , the error of one-step generation of \mathbf{x} from any s is bounded as follows:*

$$\sup_{s, \mathbf{x}} \|f_\theta(\mathbf{x}_{s/\Delta t}, \mathbf{g}, s/\Delta t) - \mathbf{x}\|_2 = O(\Delta t). \quad (6)$$

Thus, the error of one-step generation remains bounded by the global discretization error $O(\Delta t)$ in multi-step reverse process, validating that acceleration will not amplify trajectory deviation or disrupt the recommenders' effectiveness. The full proof is provided in Appendix A.1.

4.3 Fine-tuning Stage: Adaptive Preference Alignment (APA)

After improving the efficiency as detailed in Section 4.2, we further enhance diffusion-based recommenders' effectiveness by aligning the generated items with users' preferences closely, mitigating the trajectory deviation caused by the discretization error. Technically, TA-Rec introduces adaptive preference alignment (APA) in the fine-tuning stage to optimize the denoising model $f_\theta(\cdot)$ with users' preference pairs. To align the denoising trajectory with user preferences in a more refined manner, APA designs an adaptive coefficient λ_β , which controls the strength of preference optimization (as introduced in Section 3.2) adaptively. For each sequence $\mathbf{x}^{1:N-1}$, we randomly sample the negative item \mathbf{x}^- from the batch, constructing the preference pair with ground-truth next item \mathbf{x}^+ (i.e., \mathbf{x}). Since denoising from noise to oracle item is relative to user preference and the noisy degree of \mathbf{x}_t , the optimization strength λ_β can adapt at both pair- and step-wise. For a preference pair $(\mathbf{x}^+, \mathbf{x}^-)$, we define the similarity between positive and negative items as $d(\mathbf{x}^+, \mathbf{x}^-) = \text{cosine}(\mathbf{x}^+, \mathbf{x}^-)$, where cosine denotes the cosine similarity. Thus, the alignment strength $\lambda_\beta(t, s)$ can adapt according to timestep t and pair similarity $d(\mathbf{x}^+, \mathbf{x}^-)$:

$$\lambda_\beta(t, d) = \lambda_{\text{base}} \cdot \left(\left(1 - \frac{t}{T}\right) + (1 - d(\mathbf{x}^+, \mathbf{x}^-)) \right), \quad (7)$$

where λ_{base} controls the base alignment strength, we set it as $1/2$. When the pair similarity $d(\mathbf{x}^+, \mathbf{x}^-)$ and the step t is large (i.e., the preference pair is hard to distinguish and \mathbf{x}_t has a large noisy degree), the optimization strength $\lambda_\beta(t, d)$ can be small, which can avoid overfitting to ambiguous preferences and random noise. The APA loss integrates this adaptive strength $\lambda_\beta(t, d)$ into the DPO loss of Diffusion as introduced in Equation (3), and we have:

$$\mathcal{L}_{\text{APA}} = -\mathbb{E} \left[\log \sigma \left(\lambda_\beta(t, d) \log \frac{p_\theta(\mathbf{x}_t^+ | \mathbf{x}_{t+1}^+, \mathbf{g})}{p_{\text{ref}}(\mathbf{x}_t^+ | \mathbf{x}_{t+1}^+, \mathbf{g})} - \lambda_\beta(t, d) \log \frac{p_\theta(\mathbf{x}_t^- | \mathbf{x}_{t+1}^-, \mathbf{g})}{p_{\text{ref}}(\mathbf{x}_t^- | \mathbf{x}_{t+1}^-, \mathbf{g})} \right) \right], \quad (8)$$

where \mathbf{x}_t^+ and \mathbf{x}_t^- are noisy representations of \mathbf{x}^+ and \mathbf{x}^- (obtained by $q(\mathbf{x}_t|\mathbf{x}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}, (1 - \bar{\alpha}_t)\mathbf{I})$), p_{ref} denotes the reference distribution, which can be initialized using the distribution from the pretrained denoising model. The Transformer model to extract guidance signals \mathbf{g} is frozen in the fine-tuning stage. Similar to [19], the loss to optimize denoising model $f_\theta(\cdot)$ can be simplified as:

$$\begin{aligned} \mathcal{L}_{\text{APA}} = -\mathbb{E} \left[\log \sigma \left(-\lambda_\beta(t, d) \left(\|f_\theta(\mathbf{x}_t^+, \mathbf{g}, t) - \mathbf{x}^+\|_2^2 - \|f_{\text{ref}}(\mathbf{x}_t^+, \mathbf{g}, t) - \mathbf{x}^+\|_2^2 \right. \right. \right. \\ \left. \left. \left. - \left(\|f_\theta(\mathbf{x}_t^-, \mathbf{g}, t) - \mathbf{x}^-\|_2^2 - \|f_{\text{ref}}(\mathbf{x}_t^-, \mathbf{g}, t) - \mathbf{x}^-\|_2^2 \right) \right) \right) \right], \end{aligned} \quad (9)$$

where $f_{\text{ref}}(\cdot)$ is the reference model initialized by the denoising model from the pertaining stage. Below, we justify that the pair- and step-aware adaptive coefficient $\lambda_\beta(t, d)$ allows the denoising model to align the generation with user preferences more closely.

Theorem 2 *Suppose that the preference of f_θ and f_{ref} have the following relation: $f_{\text{ref}}(\mathbf{x}_t^+, \mathbf{g}, t)/f_{\text{ref}}(\mathbf{x}_t^-, \mathbf{g}, t) \leq f_\theta(\mathbf{x}_t^+, \mathbf{g}, t)/f_\theta(\mathbf{x}_t^-, \mathbf{g}, t) \leq k f_{\text{ref}}(\mathbf{x}_t^+, \mathbf{g}, t)/f_{\text{ref}}(\mathbf{x}_t^-, \mathbf{g}, t)$, where k is a constant and $k \leq e$. Then with increasing λ_β , the parameter update of θ from preference optimization becomes more aggressive, i.e., the norm of gradient $\|\nabla_\theta \mathcal{L}_{\text{Diffusion-DPO}}\| \propto \lambda_\beta$.*

Theorem 2 justifies that the denoising model’s parameter θ updates more slowly when the coefficient $\lambda_\beta(t, d)$ is small. The proof is detailed in the Appendix A.2. When the pair similarity $d(\mathbf{x}^+, \mathbf{x}^-)$ and the step t is large (i.e., the preference pair is hard to distinguish and \mathbf{x}_t has a large noisy degree), we can have a small optimization strength $\lambda_\beta(t, d)$. Thus, parameters θ of the denoising model can update more slowly, which can avoid overfitting to ambiguous preferences and random noise. And vice versa. Such adaptive strategy allows the denoising model to align the generation with user preferences more closely in a more refined manner.

4.4 Overall Pipeline

As shown in Figure 2, the pipeline of TA-Rec begins with pretraining the denoising model under Temporal Consistency Regularization (TCR) to enable efficient one-step generation. Given a user’s interaction sequence $\mathbf{x}^{1:N-1}$, we first extract guidance signals g using a Transformer encoder. The adjacent noisy representations \mathbf{x}_t and \mathbf{x}_{t-1} are generated via the forward process $q(\mathbf{x}_t|\mathbf{x}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}, (1 - \bar{\alpha}_t)\mathbf{I})$, where \mathbf{x} is the ground-truth next item. The Transformer and denoising model is then optimized using a joint loss $\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{diff}} + \lambda_c \cdot \mathcal{L}_{\text{TCR}}$. In the fine-tuning stage, Adaptive Preference Alignment (APA) fine-tunes the denoising model $f_\theta(\cdot)$ to better align generated items with user preference. For each sequence, we construct preference pairs $(\mathbf{x}^+, \mathbf{x}^-)$ by randomly sampling items that users have not interacted with as negatives. The alignment strength $\lambda_\beta(t, d)$ is dynamically adapted based on both the timestep t and pair similarity $d(\mathbf{x}^+, \mathbf{x}^-)$ as detailed in Equation (7). The APA loss \mathcal{L}_{APA} optimizes the denoising model with the guidance encoder frozen.

During inference, the model generates recommendations in a single step: given a user’s history $\mathbf{x}^{1:N-1}$, we obtain guidance g and sample noise $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ as input. Then, we generate oracle items with the denoising model: $\mathbf{x}_0 = f_\theta(\mathbf{x}_T, \mathbf{g}, T)$. The final recommendations are obtained by ranking candidate items via dot product with generated items \mathbf{x}_0 . The algorithm of the pertaining stage, fine-tuning stage, and the inference phase of TA-Rec are presented in Appendix B.

5 Experiments

In this section, we conduct extensive experiments to evaluate how TA-Rec addresses the trade-off by answering the following questions: RQ1: How does TA-Rec perform in the sequential recommendation tasks compared with leading baselines? RQ2: What are the contributions of TCR and APA in TA-Rec? RQ3: How sensitive is TA-Rec to the strength of consistency regularization and preference optimization? RQ4: How efficient is TA-Rec compared to traditional recommenders and diffusion-based recommenders? RQ5: Can TA-Rec generalize to multi-step reverse process and different pretrained diffusion-based recommenders?

5.1 Experimental Settings

Datasets. We adopt three common datasets in sequential recommendation tasks to conduct the experiments, including Yoochoose [22], KuaiRec [21], and Zhihu [20]. To process the dataset, we

exclude items with fewer than five interactions and sequences shorter than 3 interactions to mitigate cold-start issues following [4]. Then, we sort all sequences chronologically and split the data into training, validation, and testing sets in an 8:1:1 ratio to prevent data leakage. The dataset statistics are provided in Appendix C.1.

Baselines. We evaluate the performance of TA-Rec against multiple leading sequential recommenders thoroughly, including:

- Traditional Recommender: GRU4Rec [57], Caser [58], SASRec [16], Bert4Rec [59], and CL4SRec [60] predict next items by calculating the similarity between candidate items and the interaction sequences, which are modeled using GRU and Transformer architectures.
- Diffusion-based Recommender: DiffRec [6], DiffuRec [5], DreamRec [4] leverage diffusion models to formulate the adding noise and denoising process in recommenders, generating item embeddings or item scores step by step.
- Preference-based Recommender: DiffuASR [26], PDRec [17], DimeRec [61], PreferDiff [7]. DiffuASR [26], PDRec [17] employ diffusion models to generate augmented items that enrich user preference representations. Meanwhile, DimeRec [61] and PreferDiff [7] directly generate recommended items using diffusion models, enhanced by multi-interest extraction and preference pair optimization, respectively.

The detailed explanation for baselines is presented in the Appendix C.2

Implementation Details. Following DreamRec [4], the historical interaction sequence length is set to 10, with sequences containing fewer than 10 interactions padded using a padding token. Item embeddings are dimensioned at 256 for the Zhihu dataset and 64 for the KuaiRec and Yoochoose datasets. The learning rate during the pretraining stage is tuned within the range of [0.01, 0.005, 0.001, 0.0005, 0.0001, 0.00005]. The timesteps T for forward process are varied across [500, 1, 000, 2, 000]. The hyperparameter of λ_c is tuned across the range [0.1, ..., 1]. The experiments are implemented with Python 3.9 and PyTorch 2.0.1 on the Nvidia GeForce RTX 3090. We employ widely used metrics in sequential recommendation: hit ratio (HR@20) and normalized discounted cumulative gain (NDCG@20) for evaluation. Each method is tested five times, with the average performance and corresponding standard deviations reported in the tables.

5.2 Main Results (RQ1)

To answer RQ1 and validate the effectiveness of TA-Rec, we present the overall recommendation performance on three datasets in Table 1. To implement, all the methods that leverage diffusion models are based on DDPM [1] with multiple reverse steps, while PreferDiff is based on DDIM [35] for acceleration (10-20 steps) according to the original setting in [7]. Our proposed TA-Rec consistently outperforms leading baselines of three categories across all datasets, demonstrating its superiority in recommendation effectiveness. As shown in Table 1, TA-Rec achieves the best performance under both HR@20 and NDCG@20 metrics, with significant improvements over the strongest baselines (7.14%-31.82% on YooChoose, 4.65%-7.54% on KuaiRec, and 7.52%-9.64% on Zhihu). DreamRec [4] demonstrates competitive performance but incurs high computational costs due to its 1,000-step generation process. PreferDiff [7] demonstrates excellent performance by integrating BPR loss with diffusion reconstruction loss and enabling faster generation (10-20 steps) through the adoption of DDIM; however, it still trails behind TA-Rec. The superiority of our method with one-step generation validates the success of our two-stage framework in enhancing both the efficiency and effectiveness of diffusion-based recommenders.

5.3 Ablation Study (RQ2)

To validate the respective contribution of TCR and APA in TA-Rec, we conduct ablation studies with experimental results presented in Table 2. Specifically, we design several variants for TCR, where “DDPM” represents adopting the DDPM [1] paradigm to generate items step-by-step (1k steps) to recommend, guided by user interaction history. “DDIM” refers to leveraging DDIM [35] to accelerate the multi-step reverse process to just 10-20 steps. “w/o TCR” denotes fine-tuning denoising models from pretrained DDPM without TCR loss, “w/o APA” denotes pretraining denoising models without the fine-tuning stage. “w/o t”, “w/o d”, “w/o td” represent conduct preference alignment without

Table 1: Overall performance of different methods of sequential recommendation. The best score and the second-best score are bolded and underlined, respectively. The last row indicates the performance improvements of TA-Rec over the best-performing baseline method.

Methods	YooChoose		KuaiRec		Zhihu	
	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
GRU4Rec	3.89±0.11	1.62±0.02	3.32±0.11	1.23±0.08	1.78±0.12	0.67±0.03
Caser	4.06±0.12	1.88±0.09	2.88±0.19	1.07±0.07	1.57±0.05	0.59±0.01
SASRec	3.79±0.03	1.71±0.03	4.02±0.09	1.79±0.10	1.85±0.01	0.77±0.03
Bert4Rec	4.96±0.05	2.05±0.03	3.77±0.09	1.73±0.04	2.01±0.06	0.72±0.04
CL4SRec	4.67±0.03	2.12±0.01	4.43±0.07	2.66±0.03	2.11±0.05	0.76±0.04
DiffRec	4.33±0.02	1.84±0.01	3.74±0.08	1.77±0.05	1.82±0.03	0.65±0.09
DiffuRec	4.63±0.03	2.23±0.04	4.51±0.02	3.40±0.06	2.23±0.09	0.81±0.05
DreamRec	4.78±0.06	2.23±0.02	<u>5.16</u> ±0.05	<u>4.11</u> ±0.02	<u>2.26</u> ±0.07	0.79±0.01
DiffuASR	4.48±0.03	1.92±0.02	4.53±0.02	3.30±0.03	2.05±0.02	0.71±0.02
PDRec	5.43±0.02	<u>3.08</u> ±0.02	4.48±0.02	3.68±0.03	2.12±0.04	0.76±0.03
DimeRec	5.33±0.03	3.86±0.08	4.17±0.04	3.64±0.05	2.06±0.02	0.78±0.06
PreferDiff	<u>5.74</u> ±0.07	3.07±0.08	4.87±0.02	3.81±0.07	2.22±0.03	<u>0.83</u> ±0.01
Ours	6.15 ±0.03	4.06 ±0.02	5.40 ±0.02	4.42 ±0.04	2.43 ±0.01	0.91 ±0.06
improv.	7.14%	31.82%	4.65%	7.54%	7.52%	9.64%

Table 2: Ablation Study for TCR and APA. The best performance is bolded.

Methods	YooChoose		KuaiRec		Zhihu	
	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
DDPM	5.68±0.05	3.81±0.03	5.23±0.02	4.23±0.06	2.33±0.04	0.86±0.03
DDIM	5.34±0.04	3.75±0.03	5.05±0.02	3.91±0.05	2.38±0.01	0.81±0.05
w/o TCR	5.72±0.02	3.82±0.03	5.24±0.02	4.23±0.04	2.39±0.05	0.87±0.01
w/o APA	5.89±0.02	3.82±0.05	5.33±0.06	4.30±0.02	2.41±0.04	0.89±0.05
w/o td	6.10±0.02	4.04±0.04	5.36±0.04	4.39±0.05	2.28±0.05	0.87±0.02
w/o d	6.11±0.01	4.05±0.04	5.38±0.03	4.40±0.03	2.38±0.01	0.89±0.02
w/o t	6.12±0.03	4.04±0.01	5.39±0.02	4.41±0.03	2.35±0.01	0.89±0.04
Ours	6.15 ±0.02	4.06 ±0.04	5.40 ±0.02	4.42 ±0.01	2.43 ±0.04	0.91 ±0.02

considering step t or pair similarity s or both. As shown in Table 2, all the variants surpass the DDPM- or DDIM-based recommenders, demonstrating the significance of TCR and APA. Additionally, the superiority of “w/o d”, “w/o t” over “w/o td” validates the necessity of pair- and step-wise adaptation for APA. More analysis for our ablation study is detailed in Appendix D.1.

5.4 Sensitivity Analysis (RQ3)

We investigate the sensitivity of TA-Rec to the hyperparameters λ_c (controls the weight of TCR loss in pertaining stage) and λ_β (controls the strength of preference alignment in fine-tuning stage). The results of λ_c are shown in Figure 3, where “onestep” and “multistep” refer to TA-Rec using one-step generation and its extension in a multi-step setting (5-10). We can observe that the performance peaks when the regularization strength $\lambda_c \in [0.4, 0.8]$, balancing consistency smoothing and item reconstruction. Excessive regularization ($\lambda_c > 0.8$) suppresses oracle item modeling, while weak regularization ($\lambda_c < 0.3$) fails to mitigate discretization errors. The results for λ_β are presented in Figure 4, where “fix_one”, “fix_mul”, “Ours_one”, and “fix_one” refer to fine-tuning TA-Rec with fixed λ_β or our adaptive $\lambda_\beta(t, d)$ with one or multiple steps-generation, respectively. The curves of “fix_one” and “fix_mul” fluctuate as λ_β changes, highlighting the significance of λ_β in performance optimization. The curves of “Ours_one” and “Ours_mul” lie above those of “fix_one” and “fix_mul”, validating the superiority of our design of adaptive coefficient $\lambda_\beta(t, d)$ in aligning user preference.

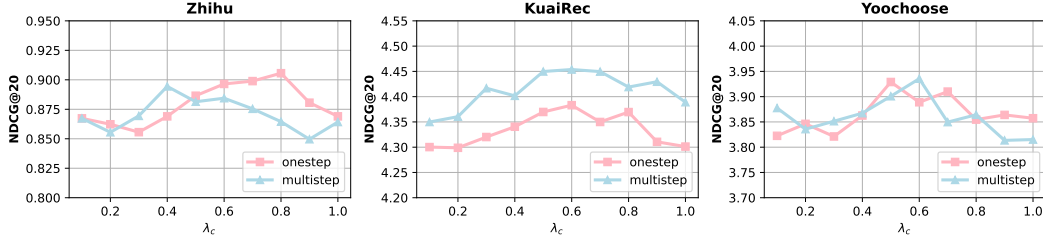


Figure 3: Performance of TA-Rec on different λ_c , demonstrating the sensitivity of TA-Rec to the strength of consistency regularization.

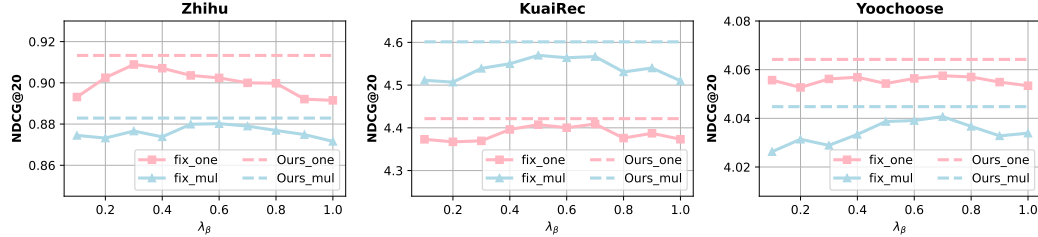


Figure 4: Performance of TA-Rec on different λ_β , demonstrating the sensitivity of TA-Rec on the strength of preference alignment and the superiority of our Adaptive preference alignment.

5.5 Computational Resource Comparison (RQ4)

To answer RQ4, we compare the training and inference efficiency of TA-Rec with traditional recommenders (*e.g.*, SASRec [16]) and diffusion-based recommenders (DreamRec [4] with 1k steps and PreferDiff [7] with 10-20 steps). The time costs are presented in Table 3. We observe that the computational complexity of TA-Rec for training each epoch is comparable to that of other diffusion-based and traditional recommenders using the same sequence encoder (*i.e.*, Transformer [62]). Moreover, by employing TCR to realize one-step generation, we significantly enhance the efficiency of TA-Rec during the inference phase. As a result, TA-Rec substantially reduces inference time compared to DreamRec and PreferDiff, achieving performance levels similar to SASRec.

Table 3: Running time comparison of TA-Rec and other recommenders that use the same sequence encoder on three datasets.

Methods	YooChoose		KuaiRec		Zhihu	
	Traning	Inferencing	Training	Inferencing	Training	Inferencing
SASRec	01m18s	00m 06s	02m 07s	00m 08s	00m 12s	00m 01s
DreamRec	01m 19s	23m 14s	02m 23s	28m 02s	00m 12s	05m 04s
PreferDiff	01m 18s	00m 14s	02m 22s	00m 32s	00m 11s	00m 03s
Ours	01m 18s	00m 06s	02m 21s	00m 07s	00m 12s	00m 01s

The analysis and experiments conducted for RQ5 are detailed in Appendix D.2, demonstrating that TA-Rec generalizes effectively in multi-step settings (1-5 steps) and that APA can enhance the performance of other diffusion-based recommenders using DDPM [1] or DDIM [35] backbones.

6 Conclusion

In this paper, we propose a novel two-stage framework, TA-Rec, to address the critical efficiency-effectiveness trade-off in diffusion-based sequential recommenders. To improve efficiency without compromising recommendation performance, TA-Rec integrates Temporal Consistency Regularization (TCR) in the pretraining stage, smoothing the denoising function to realize one-step generation with bounded error. To further enhance effectiveness, TA-Rec fine-tunes the denoising model with Adaptive Preference Alignment (APA) at both pair- and step-wise. Theoretical justification

and extensive experiments validate that TA-Rec can enhance both efficiency and effectiveness of Diffusion-based recommenders, addressing the discretization error-induced trade-off.

7 Acknowledgement

This research is supported by the National Natural Science Foundation of China (62572449, 62302321). This research is also supported by the Fundamental Research Funds for the Central Universities (WK2100250065) and the advanced computing resources provided by the Supercomputing Center of the USTC.

References

- [1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.
- [2] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*. OpenReview.net, 2021.
- [3] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, 2022.
- [4] Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. In *NeurIPS*, 2023.
- [5] Zihao Li, Aixin Sun, and Chenliang Li. Diffurec: A diffusion model for sequential recommendation. *ACM Trans. Inf. Syst.*, 42(3):66:1–66:28, 2024.
- [6] Wenjie Wang, Yiyang Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. Diffusion recommender model. In *SIGIR*, pages 832–841. ACM, 2023.
- [7] Shuo Liu, An Zhang, Guoqing Hu, Hong Qian, and Tat-Seng Chua. Preference diffusion for recommendation. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [8] Haoyang Liu, Yufei Kuang, Jie Wang, Xijun Li, Yongdong Zhang, and Feng Wu. Promoting generalization for exact solvers via adversarial instance augmentation, 2023.
- [9] Haoyang Liu, Jie Wang, Wanbo Zhang, Zijie Geng, Yufei Kuang, Xijun Li, Bin Li, Yongdong Zhang, and Feng Wu. MILP-studio: MILP instance generation via block structure decomposition. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [10] Tianle Pu, Zijie Geng, Haoyang Liu, Shixuan Liu, Jie Wang, Li Zeng, Chao Chen, and Changjun Fan. RoME: Domain-robust mixture-of-experts for MILP solution prediction across domains. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- [11] Hongyu Liu, Haoyang Liu, Yufei Kuang, Jie Wang, and Bin Li. Deep symbolic optimization for combinatorial optimization: Accelerating node selection by discovering potential heuristics. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 2067–2075, 2024.
- [12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598, 2022.
- [13] Yangming Li and Mihaela van der Schaar. On error propagation of diffusion models. In *ICLR*. OpenReview.net, 2024.
- [14] Hongjian Liu, Qingsong Xie, Tianxiang Ye, Zhijie Deng, Chen Chen, Shixiang Tang, Xueyang Fu, Haonan Lu, and Zheng-Jun Zha. Scott: Accelerating diffusion models with stochastic consistency distillation. In *AAAI*, pages 5451–5459. AAAI Press, 2025.

- [15] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 32211–32252. PMLR, 2023.
- [16] Wang-Cheng Kang and Julian J. McAuley. Self-attentive sequential recommendation. In *ICDM*, pages 197–206. IEEE Computer Society, 2018.
- [17] Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Zhanhui Kang. Plug-in diffusion model for sequential recommendation. In *AAAI*, pages 8886–8894. AAAI Press, 2024.
- [18] Vien V Mai and Mikael Johansson. Stability and convergence of stochastic gradient clipping: Beyond lipschitz continuity and smoothness. In *ICML*, pages 7325–7335. PMLR, 2021.
- [19] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *CVPR*, pages 8228–8238. IEEE, 2024.
- [20] Bin Hao, Min Zhang, Weizhi Ma, Shaoyun Shi, Xinxing Yu, Houzhi Shan, Yiqun Liu, and Shaoping Ma. A large-scale rich context query and recommendation dataset in online knowledge-sharing. *CoRR*, abs/2106.06467, 2021.
- [21] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiabin Mao, and Tat-Seng Chua. KuaiRec: A fully-observed dataset and insights for evaluating recommender systems. In *CIKM*, pages 540–550. ACM, 2022.
- [22] David Ben-Shimon, Alexander Tsikinovsky, Michael Friedmann, Bracha Shapira, Lior Rokach, and Johannes Hoerle. Recsys challenge 2015 and the YOOCHOOSE dataset. In *RecSys*, pages 357–358. ACM, 2015.
- [23] Wenyu Mao, Shuchang Liu, Haoyang Liu, Haozhe Liu, Xiang Li, and Lantao Hu. Distinguished quantized guidance for diffusion-based sequence recommendation. In *WWW*, pages 425–435. ACM, 2025.
- [24] Wenyu Mao, Zhengyi Yang, Jiancan Wu, Haozhe Liu, Yancheng Yuan, Xiang Wang, and Xiangnan He. Addressing missing data issue for diffusion-based recommendation. In *SIGIR*, pages 2152–2161. ACM, 2025.
- [25] Guoqing Hu, Zhengyi Yang, Zhibo Cai, An Zhang, and Xiang Wang. Generate and instantiate what you prefer: Text-guided diffusion for sequential recommendation. *arXiv*, 2024.
- [26] Qidong Liu, Fan Yan, Xiangyu Zhao, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Feng Tian. Diffusion augmentation for sequential recommendation. In *CIKM*, pages 1576–1586. ACM, 2023.
- [27] Wenyu Mao, Shuchang Liu, Hailan Yang, Xiaobei Wang, Xiaoyu Yang, Xu Gao, Xiang Li, Lantao Hu, Han Li, Kun Gai, et al. Robust denoising neural reranker for recommender systems. *arXiv preprint arXiv:2509.18736*, 2025.
- [28] Wenyu Mao, Jiancan Wu, Haoyang Liu, Yongduo Sui, and Xiang Wang. Invariant graph learning meets information bottleneck for out-of-distribution generalization. *Frontiers of Computer Science*, 20(1):1–16, 2026.
- [29] Jujia Zhao, Wenjie Wang, Yiyang Xu, Teng Sun, Fuli Feng, and Tat-Seng Chua. Denoising diffusion recommender model. In *SIGIR*, pages 1370–1379. ACM, 2024.
- [30] Zongwei Li, Lianghao Xia, and Chao Huang. Recdiff: Diffusion model for social recommendation. In *CIKM*, pages 1346–1355. ACM, 2024.
- [31] Haoxuan Chen, Yinuo Ren, Lexing Ying, and Grant M. Rotskoff. Accelerating diffusion models with parallel sampling: Inference at sub-linear time complexity. In *NeurIPS*, 2024.
- [32] Beier Zhu, Ruoyu Wang, Tong Zhao, Hanwang Zhang, and Chi Zhang. Distilling parallel gradients for fast ODE solvers of diffusion models. *CoRR*, abs/2507.14797, 2025.

- [33] Hong Wang, Haoyang Liu, Jian Luo, Jie Wang, et al. Accelerating pde data generation via differential operator action in solution space. In *Forty-first International Conference on Machine Learning*.
- [34] Hong Wang, Jie Wang, Minghao Ma, Haoran Shao, and Haoyang Liu. Symmap: Improving computational efficiency in linear solvers through symbolic preconditioning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- [35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*. OpenReview.net, 2021.
- [36] Xijun Li, Fangzhou Zhu, Hui-Ling Zhen, Weilin Luo, Meng Lu, Yimin Huang, Zhenan Fan, Zirui Zhou, Yufei Kuang, Zhihai Wang, et al. Machine learning insides optverse ai solver: Design principles and applications. *arXiv preprint arXiv:2401.05960*, 2024.
- [37] Zhenyu Zhou, Defang Chen, Can Wang, Chun Chen, and Siwei Lyu. Simple and fast distillation of diffusion models. In *NeurIPS*, 2024.
- [38] Yuanhao Zhai, Kevin Lin, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Chung-Ching Lin, David S. Doermann, Junsong Yuan, and Lijuan Wang. Motion consistency model: Accelerating video diffusion with disentangled motion-appearance distillation. In *NeurIPS*, 2024.
- [39] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*. OpenReview.net, 2023.
- [40] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflo: One step is enough for high-quality diffusion-based text-to-image generation. In *ICLR*. OpenReview.net, 2024.
- [41] Yufei Kuang, Jie Wang, Haoyang Liu, Fangzhou Zhu, Xijun Li, Jia Zeng, HAO Jianye, Bin Li, and Feng Wu. Rethinking branching on exact combinatorial optimization solver: The first deep symbolic discovery framework. In *The Twelfth International Conference on Learning Representations*, 2024.
- [42] Liu Haoyang, Wang Jie, Cai Yuyang, Han Xiongwei, Kuang Yufei, and HAO Jianye. Optitree: Hierarchical thoughts generation with tree search for LLM optimization modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- [43] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *CoRR*, abs/2211.01095, 2022.
- [44] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *ICLR*. OpenReview.net, 2024.
- [45] Giorgio Giannone, Akash Srivastava, Ole Winther, and Faez Ahmed. Aligning optimization trajectories with diffusion models for constrained design generation. In *NeurIPS*, 2023.
- [46] Fu-Yun Wang, Yunhao Shui, Jintan Piao, Keqiang Sun, and Hongsheng Li. Diffusion-npo: Negative preference optimization for better preference aligned generation of diffusion models. In *ICLR*, 2025.
- [47] Huaisheng Zhu, Teng Xiao, and Vasant G Honavar. Dspo: Direct score preference optimization for diffusion model alignment. In *ICLR*, 2025.
- [48] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.
- [49] Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β -dpo: Direct preference optimization with dynamic β . In *NeurIPS*, 2024.
- [50] Junkang Wu, Xue Wang, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. Alphadpo: Adaptive reward margin for direct preference optimization. In *Forty-second International Conference on Machine Learning*.

- [51] Junkang Wu, Kexin Huang, Xue Wang, Jinyang Gao, Bolin Ding, Jiancan Wu, Xiangnan He, and Xiang Wang. Repo: Relu-based preference optimization. *CoRR*, abs/2503.07426, 2025.
- [52] Zander W. Blasingame and Chen Liu. Adjointdeis: Efficient gradients for diffusion models. In *NeurIPS*, 2024.
- [53] Yinuo Ren, Haoxuan Chen, Grant M. Rotskoff, and Lexing Ying. How discrete and continuous diffusion meet: Comprehensive analysis of discrete diffusion models via a stochastic integral framework. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [54] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.
- [55] Endre Süli and David F Mayers. *An introduction to numerical analysis*. Cambridge university press, 2003.
- [56] James Foster. On the convergence of adaptive approximations for stochastic differential equations. *CoRR*, abs/2311.14201, 2023.
- [57] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *ICLR (Poster)*, 2016.
- [58] Jiayi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*, pages 565–573. ACM, 2018.
- [59] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*, pages 1441–1450. ACM, 2019.
- [60] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *ICDE*, pages 1259–1273. IEEE, 2022.
- [61] Wuchao Li, Rui Huang, Haijun Zhao, Chi Liu, Kai Zheng, Qi Liu, Na Mou, Guorui Zhou, Defu Lian, Yang Song, Wentian Bao, Enyun Yu, and Wenwu Ou. Dimerec: A unified framework for enhanced sequential recommendation via generative diffusion models. In *WSDM*, pages 726–734. ACM, 2025.
- [62] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4172–4182. IEEE, 2023.
- [63] Wenyu Mao, Jiancan Wu, Weijian Chen, Chongming Gao, Xiang Wang, and Xiangnan He. Reinforced prompt personalization for recommendation with large language models. *ACM Trans. Inf. Syst.*, 43(3):72:1–72:27, 2025.
- [64] Guoqing Hu, An Zhang, Shuo Liu, Zhibo Cai, Xun Yang, and Xiang Wang. Alphafuse: Learn id embeddings for sequential recommendation in null space of language embeddings. In *SIGIR*, 2025.

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 contributions and scope of the paper are included in the abstract and Introduction.

Guidelines:

- The answer NA 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 NA 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: We have discussed the limitations of our work in Appendix E.3.

Guidelines:

- The answer NA 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 full set of assumptions and complete proofs for all theoretical results are provided in Appendix A of our paper.

Guidelines:

- The answer NA 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 introduce the implementation details of the experiment, such as the information on hardware and software in Section 5.1. Besides, we will release the code to ease the reproducibility once accepted.

Guidelines:

- The answer NA 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: We have made the data and code used in this paper public on GitHub.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide the details of the experimental settings, such as the data split, hyperparameters, etc., in the implementation details of section 5.1.

Guidelines:

- The answer NA 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: We report the 5-test results with the average performance and corresponding standard deviation reported in the experimental result tables.

Guidelines:

- The answer NA 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: We provide the details of the compute resources in the implementation details in section 5.1.

Guidelines:

- The answer NA 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: We have made sure that our paper conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA 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: We discuss the broader impacts of our framework in Appendix E.2.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [\[Yes\]](#)

Justification: There is no risk of misuse of the proposed framework, and the datasets used in our paper are open datasets.

Guidelines:

- The answer NA 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 have cited the original paper or attached the link to the existing assets used in our paper.

Guidelines:

- The answer NA 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: [Yes]

Justification: We have introduced the new assets in our implementation details of Section 5.1, and we will attach the introduction of how to run the code and the license in the code repository on Github once accepted.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Proofs

A.1 Temporal Consistency Regularization

Theorem 1 (Error Bound for One-Step Generation) *We assume that: (i) The smooth denoising function $f_\theta(\mathbf{x}_t, \mathbf{g}, t)$ satisfies Lipschitz continuity with constants $L > 0$: $\|f_\theta(\mathbf{x}_{s_1/\Delta t}, \mathbf{g}, s_1/\Delta t) - f_\theta(\mathbf{x}_{s_2/\Delta t}, \mathbf{g}, s_2/\Delta t)\|_2 \leq L|s_1 - s_2|$ for any time $s_1, s_2 \in [0, 1]$. (ii) For any step $t \in [1, \dots, T]$ and any time $s \in [t\Delta t, (t+1)\Delta t]$, there exists a constant C such that the following smooth condition holds $\|f_\theta(\mathbf{x}_{s/\Delta t}, \mathbf{g}, s/\Delta t) - f_\theta(\mathbf{x}_t, \mathbf{g}, t)\|_2 \leq C\|f_\theta(\mathbf{x}_{t+1}, \mathbf{g}, t+1) - f_\theta(\mathbf{x}_t, \mathbf{g}, t)\|_2$. Then, since we minimize L_{pre} , the error of one-step generation of \mathbf{x} from any s is bounded as follows:*

$$\sup_{s, \mathbf{x}} \|f_\theta(\mathbf{x}_{s/\Delta t}, \mathbf{g}, s/\Delta t) - \mathbf{x}\|_2 = O(\Delta t). \quad (10)$$

Proof 1 *We estimate the error*

$$\begin{aligned} \|f_\theta(\mathbf{x}_{s/\Delta t}, \mathbf{g}, s/\Delta t) - \mathbf{x}\|_2 &\leq \|f_\theta(\mathbf{x}_{s/\Delta t}, \mathbf{g}, s/\Delta t) - f_\theta(\mathbf{x}_t, \mathbf{g}, t)\|_2 + \|f_\theta(\mathbf{x}_t, \mathbf{g}, t) - \mathbf{x}\|_2 \\ &= \|f_\theta(\mathbf{x}_{s/\Delta t}, \mathbf{g}, s/\Delta t) - f_\theta(\mathbf{x}_t, \mathbf{g}, t)\|_2 + \delta. \end{aligned} \quad (11)$$

The second term $\|f_\theta(\mathbf{x}_t, \mathbf{g}, t) - \mathbf{x}\|_2$ represents denoising model's reconstruction error, bounded by the loss \mathcal{L}_{diff} . This error is inherent to denoising model, regardless of whether a multi-step or our accelerated one-step inference is used. Since we minimize \mathcal{L}_{diff} , we have $\|f_\theta(\mathbf{x}_t, \mathbf{g}, t) - \mathbf{x}\|_2 < \delta$, δ is arbitrarily small and negligible. Using the Lipschitz condition, we have

$$\begin{aligned} \|f_\theta(\mathbf{x}_{s/\Delta t}, \mathbf{g}, s/\Delta t) - f_\theta(\mathbf{x}_t, \mathbf{g}, t)\|_2 &\leq C\|f_\theta(\mathbf{x}_{t+1}, \mathbf{g}, t+1) - f_\theta(\mathbf{x}_t, \mathbf{g}, t)\|_2 \\ &\leq LC|(t+1)\Delta t - t\Delta t| \\ &= O(\Delta t). \end{aligned} \quad (12)$$

Thus, we have $\sup_{s, \mathbf{x}} \|f_\theta(\mathbf{x}_{s/\Delta t}, \mathbf{g}, s/\Delta t) - \mathbf{x}\|_2 = O(\Delta t)$.

A.2 Adaptive Preference Alignment

Theorem 2 *Suppose that the preference of f_θ and f_{ref} have the following relation: s.t. $f_{ref}(\mathbf{x}_t^+, \mathbf{g}, t)/f_{ref}(\mathbf{x}_t^-, \mathbf{g}, t) \leq f_\theta(\mathbf{x}_t^+, \mathbf{g}, t)/f_\theta(\mathbf{x}_t^-, \mathbf{g}, t) \leq k f_{ref}(\mathbf{x}_t^+, \mathbf{g}, t)/f_{ref}(\mathbf{x}_t^-, \mathbf{g}, t)$, where k is a constant and $k \leq e$. Then with increasing λ_β , the parameter update of θ from preference optimization becomes more aggressive, i.e., the norm of gradient $\|\nabla_\theta \mathcal{L}_{Diffusion-DPO}\| \propto \lambda_\beta$.*

Proof 2 *The gradient of $\mathcal{L}_{Diffusion-DPO}$ is defined as:*

$$\|\nabla_\theta \mathcal{L}_{Diffusion-DPO}\| = \frac{1}{1 + e^{\lambda_\beta \Delta r}} \nabla_\theta(\Delta r), \quad (13)$$

where $\Delta r = \log \frac{f_\theta(\mathbf{x}_t^+, \mathbf{g}, t)}{f_{ref}(\mathbf{x}_t^+, \mathbf{g}, t)} - \log \frac{f_\theta(\mathbf{x}_t^-, \mathbf{g}, t)}{f_{ref}(\mathbf{x}_t^-, \mathbf{g}, t)}$

The gradient of $\nabla_\theta \mathcal{L}_{Diffusion-DPO}$ with respect to λ_β is:

$$\nabla_{\lambda_\beta} \nabla_\theta \mathcal{L}_{Diffusion-DPO} = \frac{1 + e^{\lambda_\beta \Delta r} - \lambda_\beta \Delta r e^{\lambda_\beta \Delta r}}{(1 + e^{\lambda_\beta \Delta r})^2} \nabla_\theta(\Delta r). \quad (14)$$

Given the initial condition:

$$\Delta r = \log \left(\frac{f_\theta(\mathbf{x}_t^+, \mathbf{g}, t)/f_\theta(\mathbf{x}_t^-, \mathbf{g}, t)}{f_{ref}(\mathbf{x}_t^+, \mathbf{g}, t)/f_{ref}(\mathbf{x}_t^-, \mathbf{g}, t)} \right) \leq \log k \leq 1. \quad (15)$$

Thus, we have:

$$\frac{1 + e^{\lambda_\beta \Delta r} - \lambda_\beta \Delta r e^{\lambda_\beta \Delta r}}{(1 + e^{\lambda_\beta \Delta r})^2} \nabla_\theta(\Delta r) \geq 0. \quad (16)$$

This implies two cases:

1. If $\nabla_\theta \mathcal{L}_{Diffusion-DPO} \geq 0$, then $\nabla_\theta(\Delta r) \geq 0$ and $\nabla_{\lambda_\beta} \nabla_\theta \mathcal{L}_{Diffusion-DPO} \geq 0$.
2. If $\nabla_\theta \mathcal{L}_{Diffusion-DPO} \leq 0$, then $\nabla_\theta(\Delta r) \leq 0$ and $\nabla_{\lambda_\beta} \nabla_\theta \mathcal{L}_{Diffusion-DPO} \leq 0$.

That means the norm of gradient $\|\nabla_\theta \mathcal{L}_{Diffusion-DPO}\| \propto \lambda_\beta$. The parameters of θ update slowly when the coefficient λ_β is small.

B Algorithm

Here we list the algorithm of TA-Rec’s pretraining, fine-tuning, and inference phase in Algorithm 1, Algorithm 2, and Algorithm 3.

Algorithm 1: Pretraining stage of TA-Rec

Input: Interaction sequence $\mathbf{x}^{1:N-1}$, next item \mathbf{x} , hyperparameters λ_c , initialized denoising model $f_\theta(\cdot)$ and Transformer encoder.
Output: optimized denoising model $f_\theta(\cdot)$.
1: $\mathbf{g} = \text{Transformer}(\mathbf{x}^{1:N-1})$ Encode the guidance.
2: $t \sim [1, \dots, T]$ ▷ Sample diffusion step.
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Sample Gaussian noise.
4: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x} + (1 - \bar{\alpha}_t) \mathbf{z}$ ▷ Add t steps Gaussian noise.
5: $\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \mathbf{x} + (1 - \bar{\alpha}_{t-1}) \mathbf{z}$ ▷ Add $t - 1$ steps Gaussian noise.
6: $\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{x}, \mathbf{g}, t} [\|f_\theta(\mathbf{x}_t, \mathbf{g}, t) - \mathbf{x}\|_2^2]$ ▷ Calculate reconstruction loss.
7: $\mathcal{L}_{\text{TCR}} = \mathbb{E}_{\mathbf{x}_t, \mathbf{g}, t} [\|f_\theta(\mathbf{x}_t, \mathbf{g}, t) - f_\theta(\mathbf{x}_{t-1}, \mathbf{g}, t-1)\|_2^2]$ ▷ Calculate TCR loss.
8: $\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{diff}} + \lambda_c \cdot \mathcal{L}_{\text{TCR}}$ ▷ Total loss of pretraining stage
9: Update $f_\theta(\cdot)$ and Transformer encoder with \mathcal{L}_{TCR} .

Algorithm 2: Finetuning stage of TA-Rec

Input: preference pair $(\mathbf{x}^+, \mathbf{x}^-)$, pretrained denoising model $f_\theta(\cdot)$, guidance \mathbf{g} .
Output: optimized denoising model $f_\theta(\cdot)$.
1: $t \sim [1, \dots, T]$ ▷ Sample diffusion step.
2: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Sample Gaussian noise.
3: $\mathbf{x}_t^+ = \sqrt{\bar{\alpha}_t} \mathbf{x}^+ + (1 - \bar{\alpha}_t) \mathbf{z}$ ▷ Add Gaussian noise to positive items.
4: $\mathbf{x}_t^- = \sqrt{\bar{\alpha}_t} \mathbf{x}^- + (1 - \bar{\alpha}_t) \mathbf{z}$ ▷ Add Gaussian noise to negative items.
5: $\lambda_\beta(t, d) = \lambda_{\text{base}} \cdot ((1 - \frac{t}{T}) + (1 - d(\mathbf{x}^+, \mathbf{x}^-)))$ ▷ Calculate adaptive coefficient.
6: $\mathcal{L}_{\text{APA}} \leftarrow \text{Equation 9}$ ▷ Adaptive preference alignment loss.
7: Update $f_\theta(\cdot)$ with \mathcal{L}_{APA}

Algorithm 3: Inference phase of TA-Rec

Input: Interaction sequence $\mathbf{x}^{1:N-1}$, optimal denoise model $f_\theta(\cdot)$, Transformer encoder.
Output: Oracle item embedding \mathbf{x}_0 .
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Sample Gaussian noise.
2: $\mathbf{g} = \text{Transformer}(\mathbf{x}^{1:N-1})$ ▷ Encode interaction sequence as guidance.
 # One-step generation
3: $\mathbf{x}_0 = f_\theta(\mathbf{x}_T, \mathbf{g}, T)$
 # Multi-step generation
4: **for** $t = T, \dots, 1$ **do**
5: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
6: $\hat{\mathbf{x}}_0 = f_\theta(\mathbf{x}_t, \mathbf{g}, t)$
7: $\mathbf{x}_{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1 - \bar{\alpha}_t} \hat{\mathbf{x}}_0 + \frac{\sqrt{\bar{\alpha}_t(1 - \bar{\alpha}_{t-1})}}{1 - \bar{\alpha}_t} \mathbf{x}_t + \sqrt{\beta_t} \mathbf{z}$ ▷ Reverse step by step.
8: **end for**
9: **return** \mathbf{x}_0

C Detailed Experimental Settings

C.1 Details of Datasets

Here, we present the statistics of our datasets in Table 4.

Table 4: Statistics of the three datasets.

Dataset	YooChoose	KuaiRec	Zhihu
#sequences	128,468	92,090	11,714
#items	9,514	7,261	4,838
#interactions	539,436	737,163	77,712

C.2 Detailed Baselines

Here, we detailed our baseline methods, including traditional Recommender, diffusion-based Recommender, and preference-based Recommender. **Traditional Recommender:** These methods predict next items by calculating the similarity between candidate items and the interaction sequences, which are modeled using GRU and Transformer architectures.

- GRU4Rec [57] uses gated recurrent units (GRUs) to model user behavior sequences, processing session data sequentially to predict next-item interactions.
- Caser [58] applies horizontal and vertical convolutional filters to capture both point-wise and union-level sequential patterns from user interaction sequences.
- SASRec [16] employs self-attention mechanisms to weight historical interactions dynamically, modeling long-range dependencies in user behavior sequences.
- Bert4Rec [59] adapts the bidirectional Transformer architecture with masked item prediction to learn contextual representations of user behavior sequences.
- CL4SRec [60] incorporates contrastive learning by constructing augmented sequence views through item cropping, masking, and reordering operations.

Diffusion-based Recommender: These methods leverage diffusion models to formulate the adding noise and denoising process in generative recommenders, generating item embeddings or item scores step by step.

- DiffRec [6] optimizes the diffusion model to predict noise in corrupted interactions, where diffusion steps progressively refine interaction probabilities.
- DiffuRec [5] employs diffusion models for sequential recommendation, which adds Gaussian noise to next items while preserving historical sequences.
- DreamRec [4] reformulates sequential recommendation as oracle item generation via classifier-free guidance-based diffusion, where encoded historical interactions serve as condition signals.

Preference-based Recommender: These methods employ diffusion models to enrich user preference representations or enhance diffusion models’ ability with learned user preferences.

- DiffuASR [26] enhances sequential recommendation by generating high-quality pseudo sequences using a diffusion model, specifically designed to better capture and align with user preferences.
- PDRec [17] leverages diffusion models to generate comprehensive user preferences on all items, using strategies like historical behavior reweighting and noise-free negative sampling to enhance the representation of user preferences.
- DimeRec [61] shifts the recommendation task from generating specific items to generating user interests, using a multi-interest model to extract stable user preferences and a diffusion model to reconstruct user embeddings.
- PreferDiff [7] introduces a tailored optimization objective for diffusion-based recommenders, transforming BPR into a log-likelihood ranking objective and integrating multiple negative samples to better capture and align with user preferences

Table 5: Ablation Study for Temporal Consistency Regularization. The best performance is bolded.

Methods	YooChoose		KuaiRec		Zhihu	
	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
DDPM	5.68 \pm 0.05	3.81 \pm 0.03	5.23 \pm 0.02	4.23 \pm 0.06	2.33 \pm 0.04	0.86 \pm 0.03
DDIM	5.34 \pm 0.04	3.75 \pm 0.03	5.05 \pm 0.02	3.91 \pm 0.05	2.38 \pm 0.01	0.81 \pm 0.05
w/o TCR	5.72 \pm 0.02	3.82 \pm 0.03	5.24 \pm 0.02	4.23 \pm 0.04	2.39 \pm 0.05	0.87 \pm 0.01
w/ TCR	5.89 \pm 0.02	3.82 \pm 0.05	5.33 \pm 0.06	4.30 \pm 0.02	2.41 \pm 0.04	0.89 \pm 0.05
Ours	6.15 \pm 0.02	4.06 \pm 0.04	5.40 \pm 0.02	4.42 \pm 0.01	2.43 \pm 0.04	0.91 \pm 0.02

Table 6: Ablation Study for the adaptive strategies for preference Alignment.

Methods	YooChoose		KuaiRec		Zhihu	
	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
w/o APA	5.89 \pm 0.02	3.82 \pm 0.05	5.33 \pm 0.06	4.30 \pm 0.02	2.41 \pm 0.04	0.89 \pm 0.05
w/o td	6.10 \pm 0.02	4.04 \pm 0.04	5.36 \pm 0.04	4.39 \pm 0.05	2.28 \pm 0.05	0.87 \pm 0.02
w/o d	6.11 \pm 0.01	4.05 \pm 0.04	5.38 \pm 0.03	4.40 \pm 0.03	2.38 \pm 0.01	0.89 \pm 0.02
w/o t	6.12 \pm 0.03	4.04 \pm 0.01	5.39 \pm 0.02	4.41 \pm 0.03	2.35 \pm 0.01	0.89 \pm 0.04
Ours	6.15 \pm 0.02	4.06 \pm 0.04	5.40 \pm 0.02	4.42 \pm 0.01	2.43 \pm 0.04	0.91 \pm 0.02

D More Experimental Results

D.1 Detailed Analysis for Ablation Study

We present the ablation study for TCR and APA in Tables 5 and 6. From the results in Table 5, we observe that “DDPM” achieves moderate performance but incurs high inference costs with a 1,000-step reverse process. In contrast, “DDIM” reduces reverse steps to 10-20 but experiences a performance drop (*e.g.*, 3.75% for “DDIM” vs. 3.81% for “DDPM” on YooChoose). However, “w/ TCR”, which employs TCR to facilitate one-step generation through a smoothed denoising function, outperforms both “DDPM” and “DDIM”, underscoring TCR’s advantage in accelerating generation without sacrificing recommendation performance. Meanwhile, “w/o TCR”, which fine-tunes DDPM-based recommendations pretrained without TCR loss, surpasses the “DDPM” variant but lags behind our TA-Rec, further highlighting the importance of TCR loss during the pretraining stage.

Additionally, as shown in Table 6, “w/o td”, “w/o d”, “w/o t”, and our TA-Rec all outperform “w/o APA” across the three datasets, confirming the effectiveness of APA in aligning user preferences during the fine-tuning stage. Removing step-wise adaptation (“w/o t”) or pair-wise adaptation (“w/o d”) degrades the performance of our TA-Rec (*e.g.*, Zhihu HR@20 drops from 2.43% to 2.35% and 2.38%, respectively), demonstrating the necessity of both adaptations for effective alignment. This further shows that dynamically adjusting the alignment strength $\lambda_\beta(t, d)$ based on both timestep t and preference pair similarity d can help mitigate overfitting to ambiguous preferences (*i.e.*, preference pairs with higher similarity d) and noise (noisy inputs with larger t).

D.2 Generalization Ability of TA-Rec

To answer RQ5, we generalize TCR to multi-step reverse process settings. The results are presented in Figure 5, where the variants “DDPM” and “DDIM” refer to the diffusion-based sequential recommenders based on DDPM and DDIM backbones, respectively, while “Ours” denotes that based on TCR training. “Ours” consistently outperforms both variants across generation settings of 1-5 steps, as indicated by the pink column’s superiority over the green and blue columns at various reverse-step settings. Furthermore, the performance of “Ours” remains stable across different reverse-step configurations, validating that TCR generalizes effectively in multi-step generation due to its smooth denoising function.

Additionally, we apply APA to various pre-trained diffusion-based recommenders, with results detailed in Table 7. This includes DDPM-based, DDIM-based, and TCR-based recommenders. For TCR-based recommenders, we evaluate outcomes from one-step and multi-step (5-step) generation.

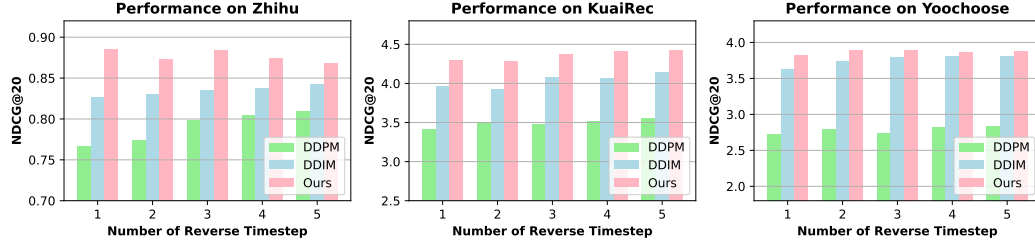


Figure 5: Performance of TA-Rec on multi-step inference settings, demonstrating the generalization ability and robustness of TCR in the multi-step reverse process.

Table 7: Performance of TA-Rec on different pretrained diffusion-based recommenders, demonstrating the generalization ability and effectiveness of APA in enhancing diffusion-based recommenders’ effectiveness.

Methods	YooChoose		KuaiRec		Zhihu	
	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
DDPM	5.68±0.05	3.81±0.03	5.23±0.02	4.23±0.06	2.33±0.04	0.86±0.03
+APA	5.72±0.03	3.82±0.01	5.24±0.03	4.24±0.05	2.39±0.02	0.87±0.04
DDIM	5.34±0.04	3.75±0.03	5.05±0.02	3.91±0.05	2.38±0.01	0.81±0.05
+APA	5.37±0.03	3.77±0.06	5.14±0.04	4.07±0.05	2.40±0.03	0.83±0.03
TCR-1	5.89±0.02	3.82±0.05	5.33±0.06	4.30±0.02	2.41±0.04	0.89±0.05
+APA	6.15±0.02	4.06±0.04	5.40±0.02	4.42±0.01	2.43±0.04	0.91±0.02
TCR-m	5.93±0.04	3.88±0.02	5.38±0.03	4.43±0.04	2.37±0.05	0.86±0.05
+APA	6.00±0.03	4.04±0.03	5.40±0.03	4.60±0.04	2.41±0.02	0.88±0.04

As shown in Table 7, applying the APA strategy to these pre-trained diffusion-based recommenders yields improved recommendation performance across all three datasets. For instance, “DDIM+APA” enhances the HR@20 performance of “DDIM” from 5.05% to 5.14% on KuaiRec. This illustrates the generalization capability of APA, which adaptively aligns denoising models with user preferences to enhance the effectiveness of diffusion-based recommenders.

D.3 Exploration of Different Negative Sampling Strategies of TA-Rec

As presented in Sec 4.4, we randomly sample one item from the item corpus that the user has not interacted with as the negative to construct a preference pair for each user. Since our main contribution is the adaptive alignment strength, which can mitigate overfitting to ambiguous preferences and noise, the random sampling strategy is employed for simplicity.

To explore the impact of different negative sampling strategies, we perform additional experiments on TA-Rec with results presented in Table 8. The “ours-hard” represents selecting hard negatives based on high cosine similarity to the positive item. The “ours-popular” denotes selecting negative items according to item popularity. The slightly superior performance of “ours-popular” and “ours-hard” over “ours” suggests that adopting different negative sampling strategies can be further considered.

D.4 Analysis on the Diversity of Diffusion-based Recommenders

For diffusion-based recommenders, the mechanism of adding random noise to target items and generating “oracle items” from stochastic noise inherently enhances the diversity of recommendation results. Our model, TA-Rec, preserves this fundamental mechanism and accelerates generation by smoothing the denoising function. The TCR loss is designed to ensure the discretization error remains bounded during this accelerated generation. It does not eliminate the model’s dependence on the random noise, so the potential for diverse generation is maintained in TA-Rec.

Table 8: Experiments on different negative sampling strategies.

Methods	YooChoose		KuaiRec		Zhihu	
	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
w/o APA	5.89	3.82	5.33	4.30	2.41	0.89
ours	6.15	4.06	5.40	4.42	2.43	0.91
ours-popular	6.17	4.07	5.43	4.43	2.43	0.92
ours-hard	6.16	4.08	5.42	4.40	2.45	0.93

Table 9: Experiments on the diversity (coverage@20) of diffusion recommenders.

Methods	YooChoose	KuaiRec	Zhihu
SASRec	0.1703	0.8368	0.7306
DreamRec	0.2051	0.8426	0.7616
ours	0.2042	0.8416	0.7609

To validate the diversity of diffusion-based recommender and our TA-Rec, we conduct additional experiments on the coverage metric, which measures the proportion of unique items recommended across multiple users. The experimental results presented in Table 9 indicate that both TA-Rec and DreamRec, which leverage diffusion models for generating recommendations, exhibit greater diversity compared to traditional recommenders like SASRec. This underscores the advantages of diffusion models in achieving diverse recommendations. Furthermore, the recommendations from TA-Rec show comparable diversity to those from DreamRec, suggesting that the inherent diversity benefits of diffusion models are maintained even with the faster inference speed.

D.5 Experiments on More Datasets.

Here, we conduct experiments to validate TA-rec on more diverse datasets (Steam [63], Beauty [64], and Toys), varying in sizes and domains. The statistics of these datasets are shown in Table 10. Experimental results are presented in Table R2.

Our method consistently outperforms various baselines on larger dataset (Steam) and diverse datasets (Amazon-beauty and Amazon-toys), further highlighting the effectiveness of TA-Rec.

E Discussion and Limitation

E.1 Relationship with Consistency models

Our Temporal Consistency Regularization (TCR) shares conceptual similarities with consistency models [15] in accelerating diffusion-based generation by enforcing self-consistency across timesteps. However, TCR differs fundamentally in its design and applicability to sequential recommendation scenarios:

Stochastic vs. Deterministic Trajectories: Consistency models typically enforce consistency along deterministic ODE trajectories, assuming a smooth and predefined path for generation. In contrast, TCR operates on the stochastic SDE process [2], which inherently models uncertainty in user behavior sequences. This is critical for recommendation systems, where user preferences are noisy and non-deterministic. By regularizing the denoising results of adjacent steps, TCR preserves the stochastic nature of denoising while smoothing the trajectory, making it more robust to the dynamic and noisy patterns of real-world user interactions.

With vs. without distillation: Consistency models often require distilling knowledge from a pre-trained diffusion model, assuming the original model has already captured the data distribution. However, sequential recommendation tasks lack unified pretrained backbone models, making the distillation unreliable. In contrast, TCR is jointly optimized with the item reconstruction loss during pretraining, directly aligning the denoising results with oracle items. This end-to-end approach is independent of numerical solvers, eliminating the need for distillation with complex solvers and ensuring the denoising accuracy under consistency regularization.

Table 10: Statistics of the Steam, Beauty, and Toys datasets.

Dataset	Steam	Beauty	Toys
#sequences	281,428	22,363	19,412
#items	13,044	12,101	11,924
#interactions	3,485,022	198,502	167,597

Table 11: The performance of TA-Rec and baseline methods on more datasets(Steam, Beauty, and Toys).

Methods	Steam		Toys		Beauty	
	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
GRU4Rec	10.13	4.21	5.54	3.45	6.46	3.48
Caser	15.12	6.42	8.53	4.21	8.35	4.21
SASRec	13.61	5.36	9.23	4.33	8.98	3.66
Bert4Rec	12.73	5.20	7.49	4.02	8.59	3.45
CL4SRec	15.06	6.12	9.09	5.08	10.18	4.85
DiffRec	15.09	6.89	9.18	5.25	10.21	5.14
DiffuRec	15.83	7.08	10.06	5.18	10.36	5.21
DreamRec	15.08	6.39	9.88	5.22	10.32	4.88
DiffuASR	15.74	6.59	9.39	5.19	10.03	5.16
PDRec	15.78	6.51	9.08	5.12	10.24	5.02
DimeRec	15.29	6.45	9.15	5.24	10.46	5.44
PreferDiff	15.92	7.12	10.18	5.34	10.69	5.38
Ours	16.25	7.36	10.87	5.81	10.99	5.54
improv.	2.07%	3.37%	6.78%	8.80%	2.81%	1.84%

E.2 Broader Impact

TA-Rec’s one-step generation framework substantially reduces the computational overhead of deploying diffusion-based recommender systems in industrial settings. By replacing multi-step denoising with a single-step process while maintaining accuracy, it enables real-time personalization for latency-critical applications such as live-streaming commerce and instant content delivery platforms. These efficiency improvements not only enhance recommendation quality but also boost user engagement and satisfaction in real-world applications.

TA-Rec’s approach to accelerating generation while preserving precision provides methodological insights for adapting diffusion models to multimodal recommendation tasks (*e.g.*, text, image, and video hybrids), thereby streamlining their adoption in next-generation AI services. This advancement bridges the gap between theoretical generative models and scalable, industry-ready solutions, particularly in scenarios requiring seamless integration of heterogeneous data types.

E.3 Limitations and Future Work

Training Cost Limitation: While TA-Rec achieves significant inference efficiency through one-step generation, its two-stage training framework (pretraining with TCR and fine-tuning with APA) requires more computational resources compared to end-to-end approaches. Additionally, the TCR necessitates executing the denoising model twice for the adjacent steps per training iteration, thereby incurring additional computational overhead during the pretraining stage.

Future Work: We will explore integrating large language model (LLM) scaling laws with diffusion-based generative recommendation frameworks to develop a diffusion LLM for generative recommendation. Specifically, we aim to investigate how LLM emergent capabilities—such as context understanding and semantic reasoning—scale with model size, data volume, and computational resources to enhance the expressive power of diffusion processes in capturing complex user-item interaction patterns.