



# Addressing Missing Data Issue for Diffusion-based Recommendation

Wenyu Mao  
University of Science and  
Technology of China  
Anhui, Hefei, China  
wenyumao2@gmail.com

Zhengyi Yang  
University of Science and  
Technology of China  
Anhui, Hefei, China  
yangzhy@mail.ustc.edu.cn

Jiancan Wu\*  
University of Science and  
Technology of China  
Anhui, Hefei, China  
wujcan@gmail.com

Haozhe Liu  
University of Science and  
Technology of China  
Anhui, Hefei, China  
liuhz0803@gmail.com

Yancheng Yuan  
The Hong Kong  
Polytechnic University  
Hong Kong, China  
yancheng.yuan@polyu.edu.hk

Xiang Wang  
University of Science and  
Technology of China  
Anhui, Hefei, China  
xiangwang1223@gmail.com

Xiangnan He\*  
University of Science and  
Technology of China  
Anhui, Hefei, China  
xiangnanhe@gmail.com

## Abstract

Diffusion models have shown significant potential in generating oracle items that best match user preference with guidance from user historical interaction sequences. However, the quality of guidance is often compromised by unpredictable missing data in observed sequence, leading to suboptimal item generation. Since missing data is uncertain in both occurrence and content, recovering it is impractical and may introduce additional errors. To tackle this challenge, we propose a novel dual-side Thompson sampling-based Diffusion Model (TDM), which simulates extra missing data in the guidance signals and allows diffusion models to handle existing missing data through extrapolation. To preserve user preference evolution in sequences despite extra missing data, we introduce Dual-side Thompson Sampling to implement simulation with two probability models, sampling by exploiting user preference from both item continuity and sequence stability. TDM strategically removes items from sequences based on dual-side Thompson sampling and treats these edited sequences as guidance for diffusion models, enhancing models' robustness to missing data through consistency regularization. Additionally, to enhance the generation efficiency, TDM is implemented under the denoising diffusion implicit models to accelerate the reverse process. Extensive experiments and theoretical analysis validate the effectiveness of TDM in addressing missing data in sequential recommendations. Our data and code is available at <https://github.com/maowenyu-11/TDM>.

## CCS Concepts

• Information systems → Recommender systems.

\* corresponding authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SIGIR '25, Padua, Italy

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-8-4007-1592-1/2025/07  
<https://doi.org/10.1145/3726302.3729890>

## Keywords

Diffusion Models, Recommender Systems, Missing Data

### ACM Reference Format:

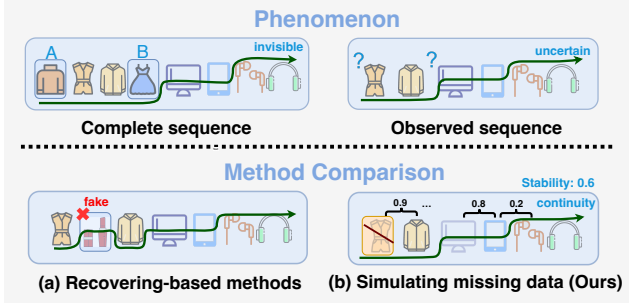
Wenyu Mao, Zhengyi Yang, Jiancan Wu\*, Haozhe Liu, Yancheng Yuan, Xiang Wang, and Xiangnan He\*. 2025. Addressing Missing Data Issue for Diffusion-based Recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*, July 13–18, 2025, Padua, Italy. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3726302.3729890>

## 1 Introduction

Sequential Recommendation [14, 15, 20, 25] is to predict the next item that aligns with a user's preferences based on his/her historical interaction sequence. Unlike conventional studies [8, 31, 46] that learn to classify target items from sampled negatives, recent studies [32, 34, 48] shift towards generating oracle items [49] with generative models that best match user preference. A promising direction is employing diffusion models [17, 28, 49], which add noise to the next items and iteratively denoise them toward oracle items, guided by interaction history conditions.

However, we argue that diffusion models' ability to generate oracle items is largely constrained by missing data in the interaction history. Typically, user interaction histories are only partially observed, with missing data occurring unpredictably [53]. Consider the case as illustrated in Figure 1, the recommender system might only observe a partial sequence, with items *A* and *B* missing due to various factors, such as privacy concerns [5] or technical limitations [47]. Consequently, diffusion models may be misled by the unreliable guidance signal from the observed sequence and generate suboptimal oracle items. Usually, missing data is **uncertain** in the observed sequence [4, 39], as it is hard to infer where the missing occurs and what content it might be due to the **invisibility** of complete sequences. Thus, leading approaches that aim to recover missing data [9, 36] and complete observed sequences [21, 50], may introduce additional errors or distort user preference accidentally, as shown in Figure 1.

To address this challenge, we propose a dual-side Thompson sampling-based Diffusion Model (TDM), which simulates extra missing data in the guidance signals rather than recovering existing one. Such simulation enables diffusion models to address missing



**Figure 1: Phenomenon of uncertain missing data in sequences and the method comparison to address it. The green curve represents the evolution of user preference over time.**

data in real-world scenarios through extrapolation, suggesting that if diffusion models can handle simulated missing data, they can also manage real missing data. To simulate while maintaining the underlying preference evolution (the changing of user preference over time) in sequences, we introduce the Dual-side Thompson Sampling [33] (DTS) strategy, which samples items to remove by exploiting known user preference. Specifically, DTS hires two probability models — one operating at the local item side and the other at the global sequence side — to capture dynamic preference:

- The local model depicts the continuity between adjacent items, reflecting shifts in user preference. As depicted in Figure 1, a high continuity score of 0.9 indicates a coherent preference for clothes, while a low score of 0.2 suggests a preference shift from mobile phones to headphones.
- The global model evaluates the stability of each entire sequence, by calculating the entropy of continuity score distribution — a high stability score indicates a stable preference, whereas a low score reflects a volatile preference. For instance, the sequence in Figure 1 experienced two significant fluctuations, resulting in a stability score of 0.6.

High-continuity items in high-stability sequences are more likely to be removed by DTS, which has little impact on the underlying preference evolution in sequences. This is evidenced by the consistent green curves between the observed sequence and the one edited by our method, as illustrated in Figure 1. Then, we treat such edited sequences as guidance for diffusion models to generate the oracle items, which can achieve consistency regularization [51] and endows diffusion models with insensitivity to preference-preserving perturbations (*i.e.*, simulated missing data).

To further improve the efficiency, we utilize denoising diffusion implicit models [35] rather than denoising diffusion probabilistic models [10] to generate oracles, which can accelerate the generation during inference. To validate, we provide a theoretical analysis of extrapolation and consistency regularization for TDM. Additionally, extensive experiments demonstrate that TDM outperforms multiple leading models in sequential recommendations. Our key contributions are as follows:

- We propose TDM to simulate extra missing data in the guidance signals, enabling diffusion models to handle existing missing data through extrapolation and consistency regularization.

- We introduce Dual-side Thompson Sampling to implement simulation based on user preference and apply denoising diffusion implicit models to accelerate generation.
- Theoretical analysis and extensive experiments validate the effectiveness of TDM in addressing missing data.

## 2 Related Work

In this section, we provide a review of missing data in recommendation and generative recommenders.

**Missing Data in Sequential Recommendation** [40, 53] refers to the absence of user behaviors in interaction sequences due to complex factors, which may lead to sequences partially observed and unreliable. To address this problem, recovering-based methods [21, 23, 50] have become mainstream, aiming to recover the complete sequence through imputation [9, 36, 42]. For instance, PDRec [23] leverages diffusion models to generate supplement items to the observed sequences. SSDRec [50] augments the interaction sequence by “insert” operations. Here we emphasize the uncertain nature [4, 39] of missing data, which poses challenges for recovering-based methods due to the invisibility of complete sequences as labels. Thus we propose simulating missing data instead of recovering it, enabling diffusion models’ extrapolation to address real missing data.

**Diffusion-based generative recommender** aims to generate oracle items that best match user preference, offering distinct advantages over discriminative recommenders that learn to classify target items from sampled negatives, particularly in sequential recommendation tasks. Since GANs and VAEs are limited in the stability and quality of generation, diffusion models have emerged as a promising technique, excelling at modeling complex data distributions and generating oracle items [24, 28, 49, 54]. For example, DiffuRec [17], DreamRec [49], and DimeRec [16] generate the next items directly by corrupting them with noise and denoising based on the historical sequence. Additionally, DiffuASR [22], DiffKG [13], and CaDiRec [2] enhance the traditional recommenders by generating sequences or items with diffusion models as data augmentation. Moreover, RecDiff [18] and DDRM [52] leverage the denoising ability of Diffusion models to improve recommenders’ robustness against noisy feedback. In our work, we emphasize the quality of diffusion models’ guidance, tackling challenges posed by uncertain missing data and enhancing the robustness through consistency regularization.

## 3 Preliminaries

In this section, we first detail denoising diffusion implicit models [35], which can accelerate the reverse process of generation. We then introduce Thompson sampling. Finally, we formulate the task of generative sequential recommendation.

### 3.1 Denoising Diffusion Implicit Models

Denoising Diffusion Implicit Models [35] is designed to generate samples faster than the original denoising diffusion probabilistic models. Here we explain its forward and reverse processes.

**Forward process:** Unlike denoising diffusion probabilistic models [10], the forward process of denoising diffusion implicit models is

not restricted as a Markovian chain, which enables it to denoise with fewer steps. Given an input data sample  $\mathbf{x}^0 \sim q(\mathbf{x}^0)$ , the forward diffusion process can be defined as:  $q(\mathbf{x}^t | \mathbf{x}^0) = \mathcal{N}(\mathbf{x}^t; \sqrt{\alpha_t} \mathbf{x}^0, (1 - \alpha_t)\mathbf{I})$ , where  $t \in [1, \dots, T]$  represents the diffusion step.  $[\alpha_1, \dots, \alpha_T]$  denotes a variance schedule. We can have:  $\mathbf{x}^t = \sqrt{\alpha_t} \mathbf{x}^0 + \sqrt{1 - \alpha_t} \epsilon$ .

**Reverse process:** Given  $\mathbf{x}^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ , denoising diffusion implicit models eliminate the noises to recover  $\mathbf{x}^0$  step by step. Formally, the reverse process from  $\mathbf{x}^t$  to  $\mathbf{x}^{t-1}$  is:

$$p_{\theta, \sigma}(\mathbf{x}^{t-1} | \mathbf{x}^t) = \mathcal{N}\left(\sqrt{\alpha_{t-1}} \hat{\mathbf{x}}^0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \frac{\mathbf{x}^t - \sqrt{\alpha_t} \hat{\mathbf{x}}^0}{\sqrt{1 - \alpha_t}}, \sigma_t^2 \mathbf{I}\right), \quad (1)$$

$$\hat{\mathbf{x}}^0 = (\mathbf{x}^t - \sqrt{1 - \alpha_t} \epsilon_{\theta}(\mathbf{x}^t)) / \sqrt{\alpha_t}. \quad (2)$$

When  $\sigma_t = 0$ , it becomes a deterministic process.  $\epsilon_{\theta}$  denotes the denoising model (e.g., U-Net [43] or Transformer [30]) parameterized by  $\theta$ , which is trained to approximate the data distribution  $q(\mathbf{x}^0)$  by maximizing the evidence lower bound of the log-likelihood  $\log p_{\theta}(\mathbf{x}^0)$ . The training loss can be derived as [35]:

$$\mathcal{L} = \sum_{t=1}^T \frac{1}{2d\sigma_t^2\alpha_t} \mathbb{E}_{\mathbf{x}^0, \epsilon} \left[ \left\| \epsilon_{\theta}(\sqrt{\alpha_t} \mathbf{x}^0 + \sqrt{1 - \alpha_t} \epsilon, t) - \epsilon \right\|_2^2 \right], \quad (3)$$

where  $\epsilon_{\theta}(\sqrt{\alpha_t} \mathbf{x}^0 + \sqrt{1 - \alpha_t} \epsilon, t)$  is the output of the denoising network to predict the noises  $\epsilon$  that add in the forward process,  $d$  is the dimension of  $\mathbf{x}^0$ .

**Acceleration:** Since the denoising objective  $\mathcal{L}$  is independent of a specific forward process as long as  $q(\mathbf{x}^t | \mathbf{x}^0)$  is fixed [35], we can redefine the non-Markovian forward process with a subsequence  $[\tau_1, \tau_2, \dots, \tau_S]$  from  $[1, \dots, T]$  as:  $q(\mathbf{x}^{\tau_s} | \mathbf{x}^0) = \mathcal{N}(\mathbf{x}^{\tau_s}; \sqrt{\alpha_{\tau_s}} \mathbf{x}^0, (1 - \alpha_{\tau_s})\mathbf{I})$ . Then, the reverse process can be reformulated as:

$$\begin{aligned} \mathbf{x}^{\tau_{s-1}} = & \sqrt{\alpha_{\tau_{s-1}}} \left( \frac{\mathbf{x}^{\tau_s} - \sqrt{1 - \alpha_{\tau_s}} \epsilon_{\theta}(\mathbf{x}^{\tau_s}, \tau_s)}{\sqrt{\alpha_{\tau_s}}} \right) \\ & + \sqrt{1 - \alpha_{\tau_{s-1}} - \sigma_{\tau_s}^2} \epsilon_{\theta}(\mathbf{x}^{\tau_s}, \tau_s) + \sigma_{\tau_s} \epsilon. \end{aligned} \quad (4)$$

With a smaller number of steps  $S$  compared to the original  $T$ , the reverse process can be accelerated.

### 3.2 Thompson Sampling

Thompson sampling [33] has emerged as a prominent exploration strategy for decision-making under uncertainty [26, 45]. To achieve a balance between exploration and exploitation [29], Thompson sampling utilizes a probability model to sample greedily based on the values of execution results from the last round. Specifically, given the value  $v$ , the probability model of Thompson sampling can be parameterized as  $F(v, p)$ , where  $p$  is a random variable that ranges from 0 to 1. A higher value of  $v$  can result in a higher sampling probability  $\hat{p}$  from the probability model. Formally, at each round, we have the sampling probability:

$$\hat{p} \sim F(v, p). \quad (5)$$

Then, Thompson sampling executes based on the sampling probability  $\hat{p}$ , updating the value  $v$  and probability model  $F(v, p)$  based on the execution results of the last round.

### 3.3 Task Formulation

For generative sequential recommendation, the goal is to generate the next item tailored to the target user conditioned on their historical interaction sequence. The mainstream solutions to this task are from the embedding perspective. Formally, we denote a user's historical interaction sequence as  $\mathbf{e}_{1:N-1} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{N-1}]$ , where  $\mathbf{e}_n$  represents the embedding of the  $n$ -th item the user has interacted with in chronological order. The subsequent item of this sequence, which we aim to generate, is represented as  $\mathbf{e}_N$ . To apply diffusion models in generative recommendation, following prior studies [28, 49], noise is first added to  $\mathbf{e}_N^0$  (equivalent to  $\mathbf{e}_N$ ), followed by a denoising process leveraging the guidance signal  $\mathbf{g}$  extracted from interaction history  $\mathbf{e}_{1:N-1}$  to ensure the generated oracle items align closely with user preferences. The core is to model the item generation distribution  $p_{\theta}(\mathbf{e}_N^{t-1} | \mathbf{e}_N^t, \mathbf{g})$  at each  $t$ -th denoising step, and inference step by step to generate the oracle items  $\mathbf{e}_N^0$ .

## 4 Methodology

In this section, we present our proposed TDM, designed to mitigate the impact of missing data, as shown in Figure 2. We begin by detailing the dual-side Thompson sampling (DTS) strategy in Section 4.1, which can simulate the mechanisms of missing data in user behaviors. Next, we describe the learning and generating phases of TDM in Section 4.2. Finally, we provide theoretical analysis for TDM in Section 4.3.

### 4.1 Dual-side Thompson Sampling

In real-world scenarios, missing data is inherently uncertain [4, 39] and hard to recover, so we simulate extra missing data in diffusion models' guidance signals, extrapolating to address the existing missing data. To preserve user preference evolution in sequences during simulation, we introduce a DTS strategy, sampling and removing items by exploiting user preference evolution with two probability models — one at the local item level and the other at the global sequence level.

**4.1.1 Definition of Two Probability Models.** User preferences often exhibit dynamic shifts between items within an interaction sequence. To capture these preference shifts locally, we introduce the concept of *continuity scores* to measure the similarity between adjacent items. Formally, given a sequence  $\mathbf{e}_{1:N-1} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{N-1}]$ , the continuity score for each item  $\mathbf{e}_n$  within the sequence is:

$$\text{con}_n = \frac{\exp(\text{sim}(\mathbf{e}_n, \mathbf{e}_{n+1}))}{\sum_{n'=1}^{N-2} \exp(\text{sim}(\mathbf{e}_{n'}, \mathbf{e}_{n'+1}))}, \quad n = 1, 2, \dots, N-2, \quad (6)$$

where  $\text{sim}(\cdot, \cdot)$  represents the cosine similarity function,  $\text{con}_n$  is the continuity score normalized using the *softmax* function. Intuitively, a higher continuity score indicates a greater similarity between adjacent items, indicating a stronger level of shared preference.

In addition to local preference shifts, user preferences often fluctuate throughout the entire sequence [19]. To assess the degree of these preference fluctuations, we calculate the entropy value  $h$  for each sequence with the continuity scores within it as below:

$$h = - \sum_{n=1}^{N-2} \text{con}_n \log(\text{con}_n). \quad (7)$$

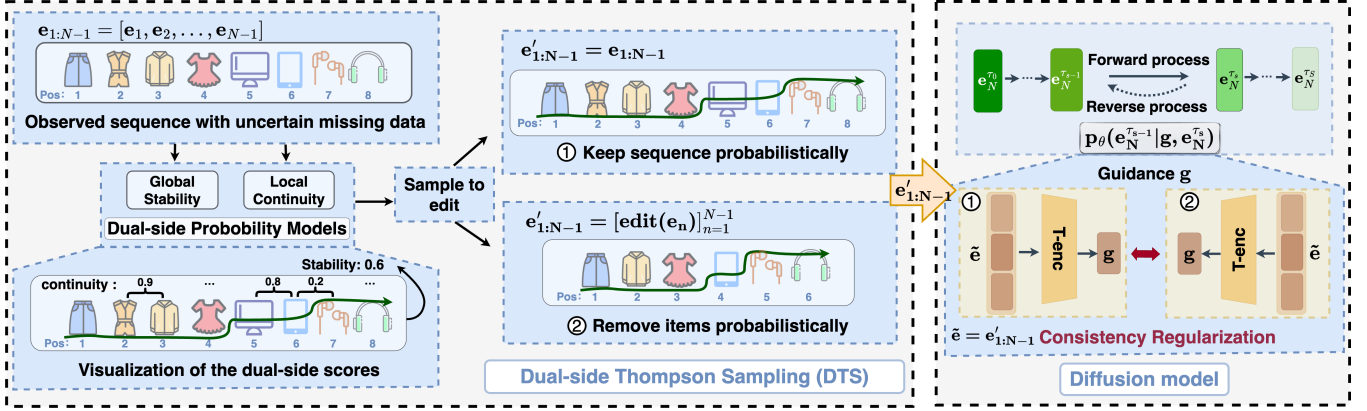


Figure 2: The overview of the TDM framework, which simulates extra missing data with DTS in the guidance signals, achieving diffusion models' consistency regularization and extrapolating to address the existing missing data.

Then, we define the *stability score*  $sta_k$  for each sequence by normalizing its entropy value  $h_k$  with softmax in a batch.

$$sta_k = \frac{\exp(h_k)}{\sum_{k'=1}^K \exp(h_{k'})}, \quad k = 1, 2, \dots, K, \quad (8)$$

where  $K$  is the number of sequences in the batch. A higher stability score reflects a higher entropy value, suggesting that user preferences remain largely unchanged throughout the sequence.

We then parameterize the two probability models with value  $con_n$  and  $sta_k$  as introduced in Section 3.2. Formally, we define the local item-side probability model as  $L(con_n, p_n)$  and the global sequence-side model as  $G(sta_k, p_k)$ , where  $p_n$  and  $p_k$  are the random variables ranging from 0 to 1. Higher values in  $con_n$  and  $sta_k$  lead to higher sampling probabilities from their respective models.

**4.1.2 Strategic Editing with DTS.** To simulate missing data while maintaining user preference evolution patterns in sequences, the DTS samples sequences to edit and items to remove based on the two probability models defined in Section 4.1.1. Formally, we have:

$$\hat{p}_n \sim L(con_n, p_n), \quad \hat{p}_k \sim G(sta_k, p_k), \quad (9)$$

where  $\hat{p}_k$  is the sampling probabilities of the  $k$ -th sequence  $e_{1:N-1}$  to be edited, and  $\hat{p}_n$  is probability for the  $n$ -th item  $e_n$  ( $1 \leq n \leq N-2$ ) within sequence  $e_{1:N-1}$  to be discarded. To preserve the last item  $e_{N-1}$  in the sequence, we manually set  $\hat{p}_{N-1} = 0$ . Since  $con_n$  and  $sta_k$  represent the local continuity and global stability respectively, the dual-side Thompson sampling strategy tends to sample items with higher continuity in sequences with greater stability scores to remove. Removing such data is expected to have little impact on the original preference shifts, as shown in the consistent green curves in Figure 2. Therefore, we can simulate uncertain data missing, while preserving the underlying preference evolution pattern.

Having established  $\hat{p}_k$  and  $\hat{p}_n$ , we can decide whether the  $n$ -th item in the  $k$ -th sequence would be discarded. Formally, the

strategically edited sequence is obtained as:

$$e'_{1:N-1} = \begin{cases} [\text{edit}(e_n)]_{n=1}^{N-1} & \text{if } 1 - \hat{p}_k < \lambda_1 \\ e_{1:N-1} & \text{otherwise} \end{cases}, \quad (10)$$

$$\text{edit}(e_n) = \begin{cases} \Phi & \text{if } 1 - \hat{p}_n < \lambda_2 \\ e_n & \text{otherwise} \end{cases}, \quad (11)$$

where  $\Phi$  is a dummy token,  $\lambda_1$  and  $\lambda_2$  are thresholds for sampling probabilities  $\hat{p}_k$  and  $\hat{p}_n$ , ranging from 0 to 1, which control the proportion of removed items. Higher values of  $\lambda_1$  and  $\lambda_2$  result in more items in more sequences being removed.

We then encode the strategically edited sequence  $e'_{1:N-1}$  as the guidance signals  $g$  using a Transformer encoder T-enc:

$$g = \text{T-enc}(e'_{1:N-1}). \quad (12)$$

In this way, the guidance is established after simulating extra missing data while preserving the evolution of users' dynamic preferences with the DTS strategy.

## 4.2 Diffusion Model for Recommendation

Having acquired the guidance  $g$  as described in Section 4.1, we then leverage  $g$  to guide diffusion models to denoise, enabling TDM to recommend items robustly in the presence of missing data. To accelerate the generation during inference, we employ denoising diffusion implicit models introduced in Section 3.1 for TDM to generate oracle items. Below, we detail TDM's training and generating phases.

**4.2.1 Training Phase.** For joint training of both conditional and unconditional models, we train TDM under the classifier-free guidance paradigm [11]. Specifically, we randomly replace the guidance  $g$  with a dummy token  $\Phi$  with probability  $\rho$ , while keeping the others unchanged. We view the next item  $e_N$  as the input  $e_N^0$  and add noise to it, following:  $e_N^{\tau_s} = \sqrt{\alpha_{\tau_s}} e_N^0 + \sqrt{1 - \alpha_{\tau_s}} \epsilon$ . Similar to DreamRec [49], we employ an MLP as the denoising neural network  $f_\theta(\cdot, \cdot, \cdot)$  to directly predict  $e_N^{\tau_s}$  into  $\hat{e}_N^0$ , rather than the noise  $\epsilon$ , guided by  $g$ :

$$\hat{e}_N^0 = f_\theta(\sqrt{\alpha_{\tau_s}} e_N^0 + \sqrt{1 - \alpha_{\tau_s}} \epsilon, g, \tau_s), \quad (13)$$

where  $\hat{\mathbf{e}}_N^0$  denotes the prediction of  $\mathbf{e}_N^0$ . According to Equation (3), the loss function of TDM can be formulated as:

$$\mathcal{L} = \sum_{s=1}^S \frac{1}{2d\sigma_{\tau_s}^2(1-\alpha_{\tau_s})} \mathbb{E}_{\mathbf{e}_N^0, \epsilon} \left[ \|\hat{\mathbf{e}}_N^0 - \mathbf{e}_N^0\|_2^2 \right]. \quad (14)$$

**4.2.2 Generating Phase.** Having trained the denoising model  $f_\theta(\cdot, \cdot, \cdot)$ , TDM can generate the oracle items step by step. Specifically, to integrate the conditional and unconditional generation under the classifier-free guidance paradigm, the denoising function is modified with a linear combination:

$$\tilde{f}_\theta(\mathbf{e}_N^{\tau_s}, \mathbf{g}, \tau_s) = (1+w)f_\theta(\mathbf{e}_N^{\tau_s}, \mathbf{g}, \tau_s) - wf_\theta(\mathbf{e}_N^{\tau_s}, \Phi, \tau_s), \quad (15)$$

where the hyperparameter  $w$  controls the guidance strength. A high value of  $w$  increases reliance on the guidance  $\mathbf{g}$ , but it may lead to overfitting. Following Equation (4), the reverse denoising step from  $\tau_s$  to  $\tau_{s-1}$  can be expressed as:

$$\begin{aligned} \mathbf{e}_N^{\tau_{s-1}} = & \sqrt{\alpha_{\tau_{s-1}}} \tilde{f}_\theta(\mathbf{e}_N^{\tau_s}, \mathbf{g}, \tau_s) \\ & + \sqrt{1-\alpha_{\tau_{s-1}}} \frac{\mathbf{e}_N^{\tau_s} - \sqrt{\alpha_{\tau_s}} \tilde{f}_\theta(\mathbf{e}_N^{\tau_s}, \mathbf{g}, \tau_s)}{\sqrt{1-\alpha_{\tau_s}}}. \end{aligned} \quad (16)$$

Under the guidance  $\mathbf{g}$  encoded from a interaction sequence, the oracle item  $\mathbf{e}_N^0$  is generated by denosing a Gaussian sample  $\mathbf{e}_N^{\tau_s} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  for  $\tau_s$  times with Equation (16). Once the oracle item is generated, we retrieve the K-nearest items from the candidate set to provide the top-K recommendation results. See <https://github.com/maowenyu-11/TDM> for the algorithms of the training and generating phases of TDM.

### 4.3 Theoretical Analysis

Here we justify TDM in simulating extra missing data, which can enable diffusion models to address existing missing data through extrapolation and consistency regularization.

**Extrapolation:** Let  $\hat{\mathbf{g}}$ ,  $\tilde{\mathbf{g}}$ , and  $\tilde{\mathbf{g}}$  denote the guidance encoded from observed sequence  $\mathbf{e}_{1:N-1}$ , unavailable complete sequence  $\mathbf{e}_{1:N-1} \oplus \delta$  and simulated sequence  $\mathbf{e}'_{1:N-1} = \mathbf{e}_{1:N-1} \ominus \delta'$ , respectively, where  $\delta$  denotes the real missing data and  $\delta'$  denotes the simulated missing data. Our objective is to demonstrate the validity of the extrapolation, specifically the inequality  $\|f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s) - f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s)\| \leq C\|f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s) - f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s)\|$  for some constant  $C$ , where  $\mathbf{e}_N^{\tau_s}$  represents the next interacted item with noise of  $\tau_s$  time steps,  $f_\theta$  is the denoising model.

Applying Taylor's Formula, we can express the two functions as follows:

$$\begin{aligned} f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s) = & f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s) + (\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s) \\ & + o(\|\tilde{\mathbf{g}} - \hat{\mathbf{g}}\|), \\ f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s) = & f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s) + (\hat{\mathbf{g}} - \tilde{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s) \\ & + o(\|\hat{\mathbf{g}} - \tilde{\mathbf{g}}\|). \end{aligned} \quad (17)$$

Then, combining the above two equalities, we have

$$\begin{aligned} & f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s) - f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s) \\ = & (\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s) + o(\|\tilde{\mathbf{g}} - \hat{\mathbf{g}}\|) \\ = & \frac{(\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s)}{(\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s)} (\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s) + o(\|\tilde{\mathbf{g}} - \hat{\mathbf{g}}\|) \\ = & \frac{(\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s)}{(\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s)} (f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s) - f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s)) \\ & + o(\|\tilde{\mathbf{g}} - \hat{\mathbf{g}}\|) + o(\|\tilde{\mathbf{g}} - \hat{\mathbf{g}}\|), \end{aligned} \quad (18)$$

where we assume that  $(\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s) \neq 0$ . Thus, we obtain the following inequality:

$$\begin{aligned} & \|f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s) - f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s)\| \\ \leq & \left| \frac{(\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s)}{(\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s)} \right| \|f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s) - f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s)\| \\ & + o(\|\tilde{\mathbf{g}} - \hat{\mathbf{g}}\|) + o(\|\tilde{\mathbf{g}} - \hat{\mathbf{g}}\|), \end{aligned} \quad (19)$$

where  $\|f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s) - f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s)\|$  is the distance between the prediction from observed sequence and simulated sequence. To bound the coefficients  $\left| \frac{(\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s)}{(\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s)} \right|$ , we need to analyze the two inner products,  $(\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s)$  and  $(\tilde{\mathbf{g}} - \hat{\mathbf{g}})^\top \nabla_{\mathbf{g}} f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s)$ .

If we can **simulate the mechanism of missing data** — specifically, if the missing data process from  $\mathbf{e}_{1:N-1}$  to  $\mathbf{e}_{1:N-1} \ominus \delta'$  can align well with that from  $\mathbf{e}_{1:N-1} \oplus \delta$  to  $\mathbf{e}_{1:N-1}$  — the difference between the two pair of data will be roughly equivalent. Consequently, the two differences in guidance,  $\tilde{\mathbf{g}} - \hat{\mathbf{g}}$  and  $\hat{\mathbf{g}} - \tilde{\mathbf{g}}$ , will also be approximately equal. In this scenario, the coefficient will be close to 1, resulting in a bounded value  $C > 0$ .

Thus we can validate that enhancing diffusion models' insensitivity to simulated missing data enables resilience against real missing data. Here, we implement the simulation mechanism as Dual-side Thompson Sampling (DTS), which preserves user preferences in sequences throughout the simulation.

**Consistency regularization:** To ensure the effectiveness of extrapolation, we leverage consistency regularization to minimize  $\|f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s) - f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s)\|$ . Since the interaction sequences are edited probabilistically across different epochs,  $\hat{\mathbf{g}}$  and  $\tilde{\mathbf{g}}$  can serve as the perturbed pairs. Let the ground-truth label of the next item be  $\mathbf{e}_N^0$ . By completing the square, we obtain the inequality:

$$\begin{aligned} & \|f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s) - f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s)\|_2^2 \\ \leq & 2 \left( \|f_\theta(\mathbf{e}_N^{\tau_s}, \hat{\mathbf{g}}, \tau_s) - \mathbf{e}_N^0\|_2^2 + \|f_\theta(\mathbf{e}_N^{\tau_s}, \tilde{\mathbf{g}}, \tau_s) - \mathbf{e}_N^0\|_2^2 \right). \end{aligned} \quad (20)$$

Consequently, we can achieve consistency regularization of minimizing the left-hand side by minimizing the right-hand side, which stems from our reconstruction loss. Such consistency regularization endows diffusion models with insensitivity to the simulated missing data, allowing the extrapolation to resist the real missing data issue.

**Table 1: Statistics of the five datasets.**

Dataset	YooChoose	KuaiRec	Zhihu	Steam	Beauty	Toys
#sequences	128,468	92,090	11,714	281,428	22,363	19,4124
#items	9,514	7,261	4,838	13,044	12,101	11,924
#interactions	539,436	737,163	77,712	3,485,022	198,502	167,597

## 5 Experiments

In this section, we conduct extensive experiments across three datasets to evaluate the effectiveness of TDM by answering the following questions. **RQ1:** How does TDM perform in the sequential recommendation compared with diverse baseline models? **RQ2:** What are the respective contributions of probability models and denoising diffusion implicit models to our method? **RQ3:** How sensitive is TDM to the thresholds of removing? **RQ4:** How robust is TDM to varying degrees of missing data in datasets and different sequence lengths? **RQ5:** Can DTS generalize on traditional recommender systems rather than diffusion models?

### 5.1 Experimental Settings

**Datasets.** We conduct experiments on three real-world datasets for sequential recommendation following the settings of DreamRec [49]: YooChoose [1], KuaiRec [6], and Zhihu [7]. To mitigate cold-start issues, we implement a preprocessing step that excludes items with fewer than five interactions and sequences shorter than 3 interactions. For each dataset, we sort all sequences chronologically and split the data into training, validation, and testing sets in an 8:1:1 ratio, ensuring that later interactions don’t leak into the training data [12]. Additionally, to validate the effectiveness of TDM on larger or diverse datasets from different domains, we also conduct experiments on Steam, Amazon-beauty, and Amazon-toys. The detailed statistics of datasets are provided in Table 1.

**Baselines.** We compare the performance of TDM against multiple leading approaches, including:

- Traditional sequential recommenders: GRU4Rec [8], Caser [38], SASRec [14], Bert4Rec [37], CL4SRec [46], IPS [44] and AdaRanker [3], which employ neural networks to model data distribution and capture user preferences.
- Generative recommenders: DiffRec [41], DiffRIS [28], and DreamRec [49], which generate target items to recommend directly with diffusion models.
- Recovering-based algorithms: DiffuASR [22], CaDiRec [2], PDRRec [23], STEAM [21], and SSDRec [50]. DiffASR, CaDiRec, and PDRRec generate supplement items to the observed sequences with diffusion models to enhance traditional sequential recommenders. STEAM and SSDRec aim to correct the interaction sequence by “insert” or other operations.

**Implementation Details.** Following DreamRec [49], we set the sequence length to 10, padding sequences with fewer than 10 interactions using a padding token. Our experiments are implemented using Python 3.9 and PyTorch 2.0.1, with computations performed on Nvidia GeForce RTX 3090 GPUs. The dimension of item embeddings is 64 across all models. The learning rate is tuned within the range of [0.01, 0.005, 0.001, 0.0005, 0.0001, 0.00005]. For diffusion

**Table 2: Overall performance of different methods for the sequential recommendation. The best score and the second-best score are bolded and underlined, respectively. The last row indicates the performance improvements of TDM over the best-performing baseline method.**

Methods	YooChoose		KuaiRec		Zhihu	
	HR(%)	NDCG(%)	HR(%)	NDCG(%)	HR(%)	NDCG(%)
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	3.68±0.08	1.63±0.02	3.92±0.18	1.53±0.11	1.62±0.01	0.60±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	4.45±0.04	1.86±0.02	4.25±0.10	2.01±0.09	2.03±0.06	0.74±0.03
IPS	3.81±0.05	1.73±0.03	3.73±0.03	1.40±0.05	1.66±0.04	0.64±0.02
AdaRanker	3.74±0.06	1.67±0.04	4.14±0.09	1.89±0.05	1.70±0.04	0.61±0.02
STEAM	4.69±0.06	1.76±0.02	4.98±0.05	2.90±0.02	1.75±0.02	0.69±0.02
SSDRec	4.52±0.07	1.95±0.03	4.19±0.08	3.28±0.06	2.03±0.06	0.72±0.03
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
CaDiRec	5.05±0.05	2.21±0.10	2.56±0.04	1.79±0.03	2.14±0.05	0.72±0.07
PDRRec	<u>6.22±0.03</u>	<u>3.17±0.02</u>	4.42±0.03	3.55±0.04	2.10±0.03	0.74±0.02
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
DiffRIS	4.51±0.03	1.95±0.02	4.28±0.03	2.03±0.04	-	-
DreamRec	4.78±0.06	2.23±0.02	<u>5.16±0.05</u>	<u>4.11±0.02</u>	<u>2.26±0.07</u>	<u>0.79±0.01</u>
TDM	<b>6.90±0.01</b>	<b>4.34±0.03</b>	<b>5.48±0.02</b>	<b>4.77±0.04</b>	<b>2.65±0.03</b>	<b>0.88±0.04</b>
Improv.	9.85%	26.95%	5.84%	13.84%	14.72%	10.23%

models, we varied the total diffusion step  $T$  across [500, 1000, 2000], employing intervals of 100 to obtain corresponding  $\tau_S$  values of [5, 10, 20] for denoising diffusion implicit models. The guidance strength  $w$  is set within the range [0, 2, 4, 6, 8, 10], and the threshold  $\lambda_1, \lambda_2$  are tuning across the range [0, 0.1, ..., 1]. We set the unconditional training probability  $p$  as 0.1 suggested by Ni et al. [27]. We adopt the widely used metrics in sequential recommendation: hit ratio (HR@20) and normalized discounted cumulative gain (NDCG@20) [14] to evaluate the recommendation performance. In our result tables, we report the average performance of five experimental runs, with their corresponding standard deviations.

### 5.2 Main Results (RQ1)

To answer RQ1, we compare the recommendation performance of TDM against multiple baselines. Table 2 and Table 3 present the experimental results on different datasets, demonstrating the superiority of TDM. For example, on the KuaiRec dataset, TDM outperforms DreamRec, a generative recommender that utilizes diffusion models guided by observed sequences, with increases of 5.84% and 13.84% in HR@20 and NDCG@20, respectively. Similarly, on the YooChoose dataset, TDM outperforms PDRRec, a recovering-based method, with improvements of 9.85% in HR@20 and 26.95% in NDCG@20. These indicate that simulating missing data with DTS enhances the robustness of diffusion models to unreliable sequences, thereby improving overall recommendation performance.

### 5.3 Ablation Study (RQ2)

The two probability models are designed to capture the dynamic evolution of user preferences, thereby preserving these evolution



**Table 3: Experimental results on larger dataset (Steam) and diverse datasets from different domains (Beauty and Toys).**

Methods	Steam		Toys		Beauty	
	HR(%)	NDCG(%)	HR(%)	NDCG(%)	HR(%)	NDCG(%)
GRU4Rec	9.23±0.05	3.56±0.03	3.18±0.08	1.27±0.03	3.85±0.09	1.38±0.06
Caser	15.20±0.09	6.62±0.05	8.83±0.09	4.02±0.05	8.67±0.06	4.36±0.10
SASRec	13.61±0.06	5.36±0.08	9.23±0.07	4.33±0.02	8.98±0.12	3.66±0.07
Bert4Rec	12.73±0.08	5.20±0.07	4.59±0.08	1.90±0.06	5.79±0.11	2.35±0.12
CL4SRec	15.06±0.08	6.12±0.06	9.09±0.03	5.08±0.03	10.18±0.11	4.85±0.12
IPS	15.65±0.08	6.46±0.02	9.29±0.01	5.27±0.04	10.15±0.02	4.56±0.07
AdaRanker	15.71±0.07	6.68±0.08	8.18±0.02	4.33±0.02	8.03±0.08	3.80±0.06
DiffuASR	15.74±0.04	6.59±0.06	9.39±0.04	5.19±0.06	10.03±0.06	5.16±0.11
CaDiRe	15.65±0.07	6.42±0.12	9.33±0.03	5.16±0.11	9.85±0.08	4.46±0.04
PDRec	15.78±0.07	6.51±0.08	9.08±0.08	5.12±0.06	10.24±0.06	5.02±0.09
DiffRec	15.09±0.04	6.89±0.03	9.18±0.06	5.25±0.04	10.21±0.04	5.14±0.02
DreamRec	15.08±0.08	6.39±0.08	9.18±0.08	5.22±0.08	10.32±0.03	4.88±0.07
TDM	16.19±0.01	7.52±0.03	9.88±0.01	5.39±0.03	10.72±0.06	5.40±0.04
Improv.	2.53%	8.38%	4.96%	2.60%	3.73%	7.04%

patterns in sequences despite the extra missing data simulated by DTS. To evaluate the impacts of the two probability models, we conduct ablation studies with eight variants of TDM. The experimental results are shown in Table 4. “w/o L” and “w/o G” indicate variants where the local or global probability model is replaced with random sampling, “w/o GL” denotes replacing both models with random sampling, “Base” represents generating oracle items without simulating missing data. To extend our method, we propose other metrics for probability models and compare their performance. Specifically, “w/P” and “w/I” denote parameterizing the local probability model with item popularity score (measured by the frequency of each item in all interactions) or score of item position in the sequence. Meanwhile, “w/D” and “w/S” represent parameterizing the global probability model with intra-sequence diversity score or score for sequence length.

As can be seen, almost all variants (*i.e.*, “w/o GL”, “w/o G”, “w/o L”, “w/P”, “w/I”, “w/D”, and “w/S”) outperform the “Base” model, highlighting the effectiveness of simulating missing data in enhancing the robustness of diffusion models, and hence improving the recommendation performance. Furthermore, our proposed TDM outperforms other variants (including “w/o L”, “w/o G”, “w/o GL”, “w/P”, “w/I”, “w/D”, and “w/S”). This demonstrates the effectiveness of continuity and stability metrics in accounting for the evolution of user preferences when generating extra missing data, which aligns more closely with the missing mechanisms discussed in the theoretical proof in Section 4.3.

Denosing diffusion implicit model (DDIM) [35] is designed to accelerate the reverse process while maintaining the comparable performance of denosing diffusion probabilistic model (DDPM) [10]. We conduct experiments to demonstrate the impact of denosing diffusion implicit models on the performance of diffusion-based recommenders. Specifically, we apply the two models to both DreamRec [49] and TDM, yielding four variants. The experimental results are presented in Table 5.

**Table 4: Ablation Study for the metrics of probability models.**

Methods	YooChoose		KuaiRec		Zhihu	
	HR(%)	NDCG(%)	HR(%)	NDCG(%)	HR(%)	NDCG(%)
Base	4.78±0.06	2.23±0.02	5.16±0.05	4.11±0.02	2.26±0.07	0.79±0.01
w/o GL	6.24±0.07	3.91±0.06	5.37±0.05	4.19±0.06	2.30±0.05	0.80±0.02
w/o L	6.41±0.06	4.26±0.05	5.44±0.03	4.63±0.02	2.34±0.03	0.86±0.02
w/o G	6.48±0.01	4.29±0.04	5.43±0.04	4.64±0.02	2.44±0.02	0.81±0.08
w/ P	6.28±0.02	4.18±0.03	5.46±0.05	4.57±0.08	2.38±0.03	0.80±0.07
w/ I	6.28±0.04	3.96±0.02	5.20±0.03	4.55±0.04	2.24±0.08	0.79±0.05
w/ D	6.26±0.04	4.20±0.06	5.45±0.02	4.52±0.02	2.29±0.02	0.81±0.03
w/ S	6.27±0.03	4.30±0.06	5.46±0.03	4.54±0.08	2.30±0.08	0.83±0.06
TDM	6.90±0.01	4.34±0.03	5.48±0.02	4.77±0.04	2.65±0.03	0.88±0.04

**Table 5: Performance comparison of different types of diffusion models. “-P” denotes using DDPM [10], while “-I” denotes using DDIM [35].**

Methods	YooChoose		KuaiRec		Zhihu	
	HR(%)	NDCG(%)	HR(%)	NDCG(%)	HR(%)	NDCG(%)
DreamRec-P	4.78±0.06	2.23±0.02	5.16±0.05	4.11±0.02	2.26±0.07	0.79±0.01
DreamRec-I	4.85±0.06	2.26±0.07	4.93±0.03	4.01±0.07	2.25±0.02	0.82±0.02
TDM-P	6.82±0.02	4.33±0.02	5.49±0.02	4.74±0.08	2.45±0.03	0.85±0.02
TDM-I	6.90±0.01	4.34±0.03	5.48±0.02	4.77±0.04	2.65±0.03	0.88±0.04

As shown in Table 5, the performance of the two types of diffusion models on sequential recommendation tasks is comparable, as evidenced by similar results for DreamRec-P and DreamRec-I. Furthermore, TDM consistently outperforms DreamRec, demonstrating the effectiveness of our DTS in enhancing the robustness of diffusion models. This indicates that while DDIM accelerates the generation process, it does not improve recommendation performance. Instead, it is the DTS strategy that facilitates performance enhancement in diffusion-based sequential recommenders by addressing missing data issues.

#### 5.4 Sensitivity Analysis (RQ3)

We examine the sensitivity of TDM to the parameters of threshold  $\lambda_1$  and  $\lambda_2$ , which represent the proportion of edited sequences and removed items when simulating data missing in TDM. The experimental results are shown in Figure 3 and 4. We can observe that the performance remains relatively stable; however, excessive or insufficient removal can result in suboptimal outcomes. This highlights the importance of choosing appropriate thresholds for DTS to simulate missing data.

#### 5.5 Robustness of TDM (RQ4)

To validate the robustness of TDM to various missing data ratios, we create synthetic datasets with missing data proportions of 10%, 20%, and 30% respectively. We compare the recommendation performance of TDM with representative baselines, including PDRec and DreamRec, across synthetic datasets. PDRec is a recovering-based method that utilizes diffusion models to enhance traditional recommenders, while DreamRec is a generative recommender where

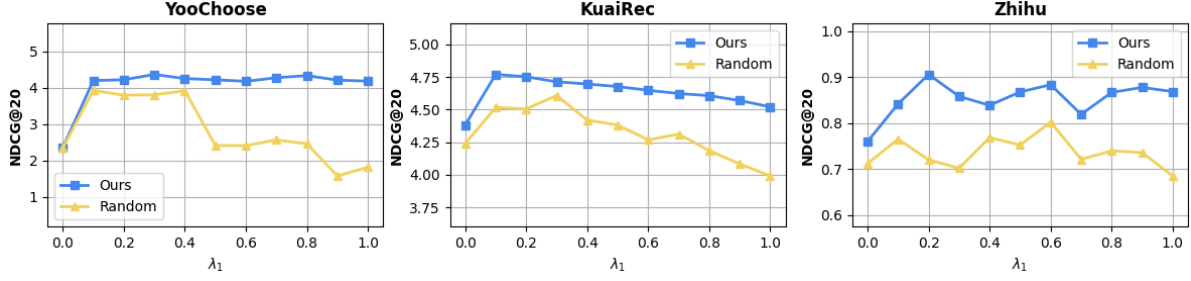


Figure 3: Sensitivity of TDM to the hyperparameter of  $\lambda_1$  on multiple datasets, demonstrating the proportion of edited sequences. The “random” represents the variant “w/o GL” of TDM.

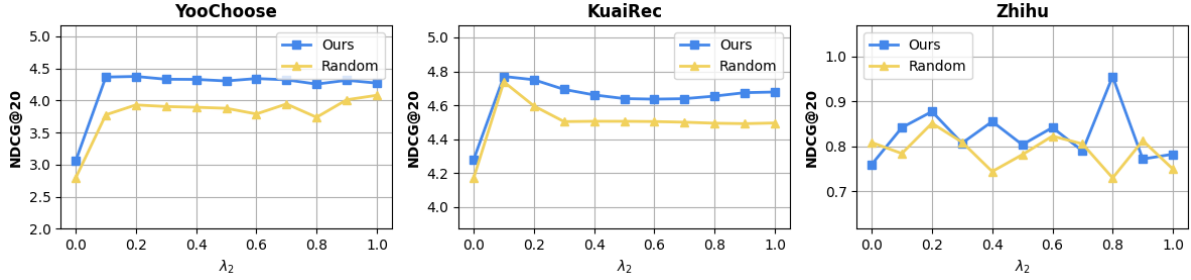


Figure 4: Sensitivity of TDM to the hyperparameter of  $\lambda_2$  on multiple datasets, demonstrating the proportion of removed items. The “random” represents the variant “w/o GL” of TDM.

diffusion models generate items to recommend directly. As shown in Figure 5, TDM outperforms these baseline models on our synthetic datasets with varying missing ratios. Furthermore, the performance decline due to increased missing data in TDM is less significant than that observed in baseline models, as evidenced by the increasing height difference between the columns. These results demonstrate the superior robustness of TDM to varying degrees of missing data in datasets, as well as the effectiveness of DTS in addressing missing data in sequential recommendation.

We further investigate the robustness of TDM on datasets with different sequence lengths (*i.e.*,  $<5$ ,  $<20$ ,  $<50$ ). We compare the recommendation performance of TDM with representative baseline models, including SASRec, Bert4Rec, and CL4SRec. The results are shown in Figure 6. TDM maintains a significant lead on different sequence lengths, validating the robustness of TDM to different sequence lengths.

## 5.6 Generalization ability of TDM (RQ5)

Introducing missing data with DTS can be considered as introducing a form of noise within the sequence, which can enhance diffusion models’ denoising ability to missing data rather than Gaussian noise. However, DTS can be **a general algorithm to enhance recommendation systems’ robustness** against uncertain missing data. To validate the performance of DTS when extended to traditional recommenders, We conduct experiments on other recommenders, including GRU4Rec, SASRec, CL4Rec, Caser, AdaRanker, and DiffuASR. The experimental results are presented in Table 6.

As shown in Table 6, applying DTS to traditional recommenders can yield nearly universal improvement in recommendation performance. This empirically proves the effectiveness and extensibility

Table 6: Comparison of TDM and traditional recommenders on performance improvement from the DTS strategy.

Methods	YooChoose		KuaiRec		Zhihu	
	HR(%)	NDCG(%)	HR(%)	NDCG(%)	HR(%)	NDCG(%)
GRU4Rec	3.89	1.62	3.32	1.23	1.78	0.67
+ DTS	3.96	1.72	3.43	1.40	1.83	0.68
Improv.	1.77%	5.81%	3.21%	12.14%	2.73%	1.47%
SASRec	3.68	1.63	3.92	1.53	1.62	0.60
+DTS	3.98	1.58	3.96	1.63	1.70	0.72
Improv.	7.54%	-3.07%	1.01%	6.13%	4.71%	11.11%
CL4SRec	4.45	1.86	4.25	2.01	2.03	0.74
+DTS	4.63	1.88	4.57	2.25	2.16	0.78
Improv.	3.89%	1.06%	7.00%	10.67%	6.02%	5.13%
Caser	4.06	1.88	2.88	1.07	1.57	0.59
+DTS	4.27	2.01	3.19	1.09	1.72	0.65
Improv.	4.92%	6.47%	9.72%	1.83%	8.72%	9.23%
AdaRanker	3.74	1.67	4.14	1.89	1.70	0.61
+DTS	4.16	1.97	4.33	1.93	1.64	0.67
Improv.	10.10%	15.22%	4.39%	2.07%	-3.53%	8.96%
DiffuASR	4.48	1.92	4.53	3.30	2.05	0.71
+DTS	4.66	2.08	4.58	3.98	2.05	0.73
Improv.	3.86%	7.69%	1.09%	17.09%	0.00%	2.82%
DreamRec	4.78	2.23	5.16	4.11	2.26	0.79
+DTS	6.90	4.34	5.48	4.77	2.65	0.88
Improv.	30.72%	48.62%	5.84%	13.84%	14.72%	10.23%

of DTS. Notably, when applied to DreamRec, a generative recommender using diffusion models, it yields the highest performance



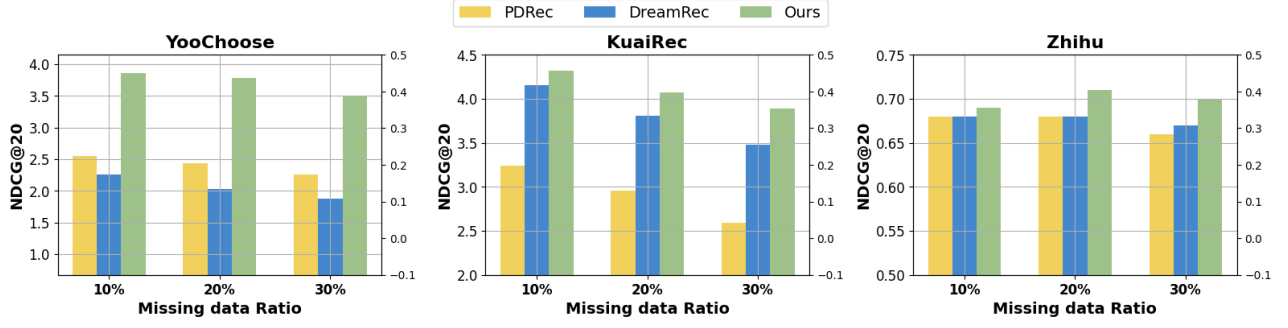


Figure 5: Performance of TDM on synthetic datasets with different missing ratios.

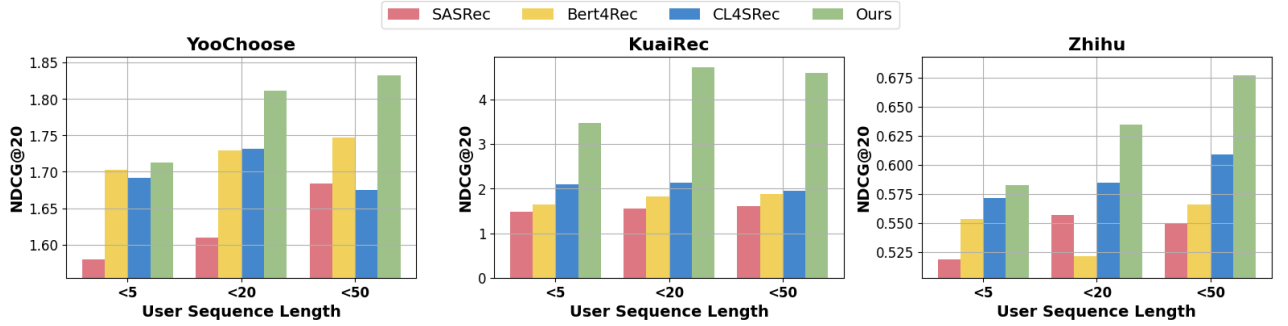


Figure 6: Performance of TDM on multiple datasets with different sequence lengths.

gains on average. This observation highlights that diffusion models provide a solid foundation for DTS to achieve consistency regularization from the empirical perspective, owing to their capability to model complex data distributions and denoise missing data.

### 5.7 Computational Resource Comparison

Since the Thompson sampling strategy relies solely on similarity computation and entropy calculation, its computational resource consumption is negligible compared to that of the transformer network architecture. As evidenced in Table 7, the computational complexity of TDM for training each epoch is nearly similar to other diffusion-based recommenders and traditional recommenders that use the same sequence encoder. Furthermore, by employing denoising diffusion implicit models to accelerate generation, we enhance the efficiency of TDM during the inference phase. As shown in Table 7, TDM substantially reduces the time cost during the inference phase than DreamRec and has a similar training time cost with other methods.

**Table 7: Running time comparison of TDM and other methods on three datasets.**

Methods	YooChoose		KuaiRec		Zhihu	
	Train	Inference	Train	Inference	Train	Inference
SASRec	01m 38s	00m 06s	02m 07s	00m 08s	00m 10s	00m 01s
AdaRanker	02m 29s	00m 08s	03m 38s	00m 09s	00m 14s	00m 01s
DreamRec	01m 31s	21m 32s	03m 59s	32m 40s	00m 14s	01m 31s
TDM	01m 22s	00m 13s	02m 23s	00m 23s	00m 11s	00m 01s

## 6 Conclusion

In this paper, we propose TDM, a novel approach that simulates extra missing data in diffusion models' guidance signals and extrapolates to address existing missing data in sequential recommendations. By introducing a dual-side Thompson sampling strategy with local and global probability models, TDM can preserve user preference evolution in sequences during simulation. Treating such edited sequences as guidance can achieve diffusion models' consistency regularization. Theoretical analysis and extensive experiments validate the effectiveness of TDM, showcasing its potential to handle missing data issues and improve recommendation performance. The limitation is that the dynamics of user preferences may exhibit intricate patterns that transcend the capabilities of our dual-side probability models. Future research could benefit from integrating a more sophisticated understanding of user preference evolution into these models. Additionally, the issue of missing data, which can stem from various factors, such as exposure bias or popularity bias, presents an opportunity for targeted simulation to enhance the robustness of diffusion models.

## 7 Acknowledgement

This research is supported by the National Natural Science Foundation of China (92270114, 62302321, 62121002). This research is also supported by the advanced computing resources provided by the Supercomputing Center of the USTC and the Research Center for Intelligent Operations Research at the Hong Kong Polytechnic University.

## References

- [1] David Ben-Shimon, Alexander Tsikinovsky, Michael Friedmann, Bracha Shapira, Lior Rokach, and Johannes Hoerle. 2015. RecSys Challenge 2015 and the YOO-CHOOSE Dataset. In *RecSys*. ACM, 357–358.
- [2] Ziqiang Cui, Haolun Wu, Bowei He, Ji Cheng, and Chen Ma. 2024. Diffusion-based Contrastive Learning for Sequential Recommendation. *CoRR* abs/2405.09369 (2024).
- [3] Xinyan Fan, Jianxun Lian, Wayne Xin Zhao, Zheng Liu, Chaozhao Li, and Xing Xie. 2022. Ada-Ranker: A Data Distribution Adaptive Ranking Paradigm for Sequential Recommendation. In *SIGIR*. ACM, 1599–1610.
- [4] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S. Yu. 2022. Sequential Recommendation via Stochastic Self-Attention. In *WWW*. ACM, 2036–2047.
- [5] Raymond Feng, Flávio Calmon, and Hao Wang. 2023. Adapting Fairness Interventions to Missing Values. In *NeurIPS*.
- [6] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022. KuaiRec: A Fully-observed Dataset and Insights for Evaluating Recommender Systems. In *CIKM*. ACM, 540–550.
- [7] Bin Hao, Min Zhang, Weizhi Ma, Shaoyun Shi, Xinxing Yu, Houzhi Shan, Yiqun Liu, and Shaoping Ma. 2021. A Large-Scale Rich Context Query and Recommendation Dataset in Online Knowledge-Sharing. *CoRR* abs/2106.06467 (2021).
- [8] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR (Poster)*.
- [9] Erna Hikmawati, Heru Nugroho, and Kridanto Surendro. 2024. Improve the Quality of Recommender Systems based on Collaborative Filtering with Missing Data Imputation. In *ICSCA*. ACM, 75–80.
- [10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *NeurIPS*.
- [11] Jonathan Ho and Tim Salimans. 2022. Classifier-Free Diffusion Guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*.
- [12] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2023. A Critical Study on Data Leakage in Recommender System Offline Evaluation. *ACM Trans. Inf. Syst.* 41, 3 (2023), 75:1–75:27.
- [13] Yangqin Jiang, Yuhao Yang, Lianghao Xia, and Chao Huang. 2024. DiffKG: Knowledge Graph Diffusion Model for Recommendation. In *WSDM*. ACM, 313–321.
- [14] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*. IEEE Computer Society, 197–206.
- [15] Xiaoyu Kong, Jiancan Wu, An Zhang, Leheng Sheng, Hui Lin, Xiang Wang, and Xiangnan He. 2024. Customizing Language Models with Instance-wise LoRA for Sequential Recommendation. In *NeurIPS*.
- [16] 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. 2025. DimeRec: A Unified Framework for Enhanced Sequential Recommendation via Generative Diffusion Models. In *WSDM*. ACM, 726–734.
- [17] Zihao Li, Aixin Sun, and Chenliang Li. 2024. DiffuRec: A Diffusion Model for Sequential Recommendation. *ACM Trans. Inf. Syst.* 42, 3 (2024), 66:1–66:28.
- [18] Zongwei Li, Lianghao Xia, and Chao Huang. 2024. RecDiff: Diffusion Model for Social Recommendation. In *CIKM*. ACM, 1346–1355.
- [19] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from History and Present: Next-item Recommendation via Discriminatively Exploiting User Behaviors. In *KDD*. ACM, 1734–1743.
- [20] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. LLaRA: Large Language-Recommendation Assistant. In *SIGIR*. ACM, 1785–1795.
- [21] Yujie Lin, Chenyang Wang, Zhumin Chen, Zhaochun Ren, Xin Xin, Qiang Yan, Maarten de Rijke, Xiuzhen Cheng, and Pengjie Ren. 2023. A Self-Correcting Sequential Recommender. In *WWW*. ACM, 1283–1293.
- [22] Qidong Liu, Fan Yan, Xiangyu Zhao, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Feng Tian. 2023. Diffusion Augmentation for Sequential Recommendation. In *CIKM*. ACM, 1576–1586.
- [23] Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Zhanhui Kang. 2024. Plug-In Diffusion Model for Sequential Recommendation. In *AAAI*. AAAI Press, 8886–8894.
- [24] Wenyu Mao, Shuchang Liu, Haoyang Liu, Haozhe Liu, Xiang Li, and Lanatao Hu. 2025. Distinguished Quantized Guidance for Diffusion-based Sequence Recommendation. *CoRR* abs/2501.17670 (2025).
- [25] Wenyu Mao, Jiancan Wu, Weijian Chen, Chongming Gao, Xiang Wang, and Xiangnan He. 2025. Reinforced prompt personalization for recommendation with large language models. *ACM Trans. Inf. Syst.* 43, 3 (2025), 1–27.
- [26] Thomas M. McDonald, Lucas Maystre, Mounia Lalmas, Daniel Russo, and Kamil Ciosek. 2023. Impatient Bandits: Optimizing Recommendations for the Long-Term Without Delay. In *KDD*. ACM, 1687–1697.
- [27] Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. In *ACL (Findings)*. Association for Computational Linguistics, 1864–1874.
- [28] Yong Niu, Xing Xing, Zhichun Jia, Ruidi Liu, Mindong Xin, and Jianfu Cui. 2024. Diffusion Recommendation with Implicit Sequence Influence. In *WWW (Companion Volume)*. ACM, 1719–1725.
- [29] Ian Osband and Benjamin Van Roy. 2015. Bootstrapped Thompson Sampling and Deep Exploration. *CoRR* abs/1507.00300 (2015).
- [30] William Peebles and Saining Xie. 2023. Scalable Diffusion Models with Transformers. In *ICCV*. IEEE, 4172–4182.
- [31] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. In *WSDM*. ACM, 813–823.
- [32] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Mahesh Sathiamoorthy. 2023. Recommender Systems with Generative Retrieval. In *NeurIPS*.
- [33] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. 2018. A Tutorial on Thompson Sampling. *Found. Trends Mach. Learn.* 11, 1 (2018), 1–96.
- [34] Naveen Sachdeva, Giuseppe Manco, Ettore Ritacco, and Vikram Pudi. 2019. Sequential Variational Autoencoders for Collaborative Filtering. In *WSDM*. ACM, 600–608.
- [35] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *ICLR*. OpenReview.net.
- [36] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *RecSys*. ACM, 213–220.
- [37] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *CIKM*. ACM, 1441–1450.
- [38] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*. ACM, 565–573.
- [39] Chenxu Wang, Fuli Feng, Yang Zhang, Qifan Wang, Xunhan Hu, and Xiangnan He. 2023. Rethinking Missing Data: Aleatoric Uncertainty-Aware Recommendation. *IEEE Trans. Big Data* 9, 6 (2023), 1607–1619.
- [40] Jun Wang, Haoxuan Li, Chi Zhang, Dongxu Liang, Enyun Yu, Wenwu Ou, and Wenjia Wang. 2023. CounterCLR: Counterfactual Contrastive Learning with Non-random Missing Data in Recommendation. In *ICDM*. IEEE, 1355–1360.
- [41] Wenjie Wang, Yiyang Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. 2023. Diffusion Recommender Model. In *SIGIR*. ACM, 832–841.
- [42] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly Robust Joint Learning for Recommendation on Data Missing Not at Random. In *ICML (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 6638–6647.
- [43] Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, and Mingyuan Zhou. 2023. Patch Diffusion: Faster and More Data-Efficient Training of Diffusion Models. In *NeurIPS*.
- [44] Zhenlei Wang, Shiqi Shen, Zhipeng Wang, Bo Chen, Xu Chen, and Ji-Rong Wen. 2022. Unbiased Sequential Recommendation with Latent Confounders. In *WWW*. ACM, 2195–2204.
- [45] Zhendong Wang and Mingyuan Zhou. 2020. Thompson Sampling via Local Uncertainty. In *ICML (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 10115–10125.
- [46] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive Learning for Sequential Recommendation. In *ICDE*. IEEE, 1259–1273.
- [47] Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. 2020. Adversarial Counterfactual Learning and Evaluation for Recommender System. In *NeurIPS*.
- [48] Yiyang Xu, Wenjie Wang, Fuli Feng, Yunshan Ma, Jizhi Zhang, and Xiangnan He. 2024. Diffusion Models for Generative Outfit Recommendation. In *SIGIR*. ACM, 1350–1359.
- [49] Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. 2023. Generate What You Prefer: Reshaping Sequential Recommendation via Guided Diffusion. In *NeurIPS*.
- [50] Chi Zhang, Qilong Han, Rui Chen, Xiangyu Zhao, Peng Tang, and Hongtao Song. 2024. SSDRec: Self-Augmented Sequence Denoising for Sequential Recommendation. In *ICDE*. IEEE, 803–815.
- [51] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. 2020. Consistency Regularization for Generative Adversarial Networks. In *ICLR*. OpenReview.net.
- [52] Jujia Zhao, Wenjie Wang, Yiyang Xu, Teng Sun, Fuli Feng, and Tat-Seng Chua. 2024. Denoising Diffusion Recommender Model. In *SIGIR*. ACM, 1370–1379.
- [53] Xiaolin Zheng, Menghan Wang, Renjun Xu, Jianmeng Li, and Yan Wang. 2022. Modeling Dynamic Missingness of Implicit Feedback for Sequential Recommendation. *IEEE Trans. Knowl. Data Eng.* 34, 1 (2022), 405–418.
- [54] Yunqin Zhu, Chao Wang, Qi Zhang, and Hui Xiong. 2024. Graph Signal Diffusion Model for Collaborative Filtering. In *SIGIR*. ACM, 1380–1390.