

# Plug-in Diffusion Model for Sequential Recommendation

Haokai Ma<sup>1</sup>, Ruobing Xie<sup>3</sup>, Lei Meng<sup>2, 1\*</sup>, Xin Chen<sup>3</sup>, Xu Zhang<sup>3</sup>, Leyu Lin<sup>3</sup>, Zhanhui Kang<sup>3</sup>

<sup>1</sup> School of Software, Shandong University, China

<sup>2</sup> Shandong Research Institute of Industrial Technology, China

<sup>3</sup> Tencent, China

mahaokai@mail.sdu.edu.cn, ruobingxie@tencent.com, lmeng@sdu.edu.cn,  
{andrewxchen, xuonezhang, goshawklin, kegokang}@tencent.com

## Abstract

Pioneering efforts have verified the effectiveness of the diffusion models in exploring the informative uncertainty for recommendation. Considering the difference between recommendation and image synthesis tasks, existing methods have undertaken tailored refinements to the diffusion and reverse process. However, these approaches typically use the highest-score item in corpus for user interest prediction, leading to the ignorance of the user’s generalized preference contained within other items, thereby remaining constrained by the data sparsity issue. To address this issue, this paper presents a novel Plug-in Diffusion Model for Recommendation (PDRec) framework, which employs the diffusion model as a flexible plugin to jointly take full advantage of the diffusion-generating user preferences on all items. Specifically, PDRec first infers the users’ dynamic preferences on all items via a time-interval diffusion model and proposes a Historical Behavior Reweighting (HBR) mechanism to identify the high-quality behaviors and suppress noisy behaviors. In addition to the observed items, PDRec proposes a Diffusion-based Positive Augmentation (DPA) strategy to leverage the top-ranked unobserved items as the potential positive samples, bringing in informative and diverse soft signals to alleviate data sparsity. To alleviate the false negative sampling issue, PDRec employs Noise-free Negative Sampling (NNS) to select stable negative samples for ensuring effective model optimization. Extensive experiments and analyses on four datasets have verified the superiority of the proposed PDRec over the state-of-the-art baselines and showcased the universality of PDRec as a flexible plugin for commonly-used sequential encoders in different recommendation scenarios. The code is available in <https://github.com/hulkima/PDRec>.

## 1 Introduction

Personalized recommendation aims to capture user preference from the massive user behaviors and predict the appropriate items the user will be interested in (Meng et al. 2020; Ma et al. 2021, 2023a). Sequential recommendation (SR) is an effective method for inferring dynamic interests from the user’s historical behavior sequences (Zhang et al. 2022; Li et al. 2022; Chen et al. 2022). However, most users in the real world only interact with a limited number of items

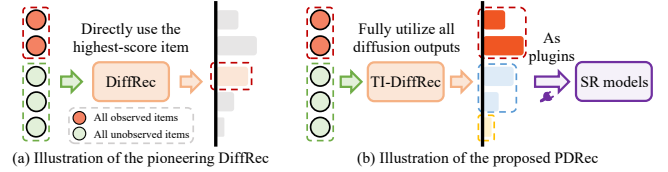


Figure 1: Illustration of the difference between the pioneering DiffRec and PDRec, where each rectangle denotes the user’s diffusion-based preference of the corresponding item.

within the overall item corpus, consequently leading to the data sparsity problem (Xia et al. 2021; Chen et al. 2023a,b).

Diffusion model (DM), benefiting from its characteristics of diverse representation and informative uncertainty, has achieved state-of-the-art results in the field of image synthesis (Ho, Jain, and Abbeel 2020), semantic segmentation (Brempong et al. 2022), and time series imputation (Lopez Alcaraz and Strodthoff 2023). This demonstrates the dominance of DM as a novel generative paradigm in multiple generation tasks. Looking back to the real-world recommender systems, it could be regarded as a generator of the complete user-item interaction matrix based on the extremely sparse supervised signals (Moon et al. 2023). It prompts an intuitive question: *Can we take full advantage of DM’s potent generalization capability to generate user preferences on both observed and unobserved items, thereby addressing the sparsity issue in recommendation?*

Recently, CODIGEM (Walker et al. 2022) and DiffRec (Wang et al. 2023) have introduced DM into recommendation, which generates users’ preferences based on their historical behaviors, yielding promising results. However, these pioneering studies still grapple with two challenges: (1) *How to fully utilize the generalized user preferences from DM?* As shown in Fig.1 (a), these methods merely utilize the highest-scored item as the final prediction in recommendation, overlooking users’ preferences towards other items in corpus and struggling with data sparsity issue. However, these preferences encapsulate substantial informative and generalized knowledge during the inference process of DM. (2) *How to incorporate the diffusion-based knowledge to construct a universal framework that could smoothly cooperate with different SR models?* These DM-based methods are primarily proposed for collaborative filtering (CF),

\*Corresponding Author.

failing to fully integrate the time-aware sequential behavioral information. This leads to huge gaps between them and real-world recommenders, limiting their practical feasibility and universality (as indicated in Table 1, even T-DiffRec exhibits notable disparities from traditional SR models).

To address these issues, we propose a novel **Plug-in Diffusion Model for Recommendation (PDRec)** framework, which leverages the diffusion models as a flexible plugin and make full use of the diffusion-generated user preferences on all items. The coarse overview of PDRec is illustrated in Fig.1 (b). Specifically, we first present a time-interval diffusion model on the basis of T-DiffRec to facilitate the more precise generation of dynamic user preferences for both observed and unobserved items. These diffusion-based preferences on all items (i.e., preferences for observed items, top-ranked unobserved items, and low-scored unobserved items) are utilized for jointly guiding the effective and stable optimization direction: (a) We devise a *Historical Behavior Reweighting (HBR)* method for users’ historical behaviors, which identifies the high-quality behaviors and reduces noisy interactions via the generated preferences given by previous diffusion models. (b) We also propose a *Diffusion-based Positive Augmentation (DPA)* approach to convert the unobserved items with top-ranked diffusion-based preferences to the potential positive labels via self-distillation, bringing in additional high-quality and diverse positive signals to alleviate data sparsity. (c) To alleviate the potential false negative sampling issue, we design a *Noise-free Negative Sampling (NNS)* strategy, which selects safer negative samples from the low-scored unobserved items provided by diffusion in training. The advantages of PDRec include: (1) HBR facilitates the discovery of more informative supervised signals from the global diffusion aspect, which could better guide model optimization. (2) DPA and NNS introduce additional knowledge on unobserved items that alleviates data sparsity issues. (3) PDRec is effective, universal, and easy-to-deploy, which could be conveniently applied to different datasets, base models, and recommendation tasks.

Extensive experiments on four real-world datasets with three base SR models demonstrate that our proposed PDRec achieves significant and consistent improvements across various datasets and tasks, including SR and cross-domain SR. Furthermore, we conduct comprehensive ablation studies and universality analyses to validate the effectiveness and universality of all components in PDRec. The main contributions of this paper are summarized as follows:

- We propose a model-agnostic Plug-in Diffusion Model for Recommendation, which fully leverages the diffusion-based preferences on all items to improve base recommenders. To the best of our knowledge, we are the first to integrate the diffusion model as a plugin for different types of recommendation models and downstream tasks.
- The proposed HBR, DPA, and NNS are effective, model-agnostic, and easy-to-deploy plug-in strategies, which involve informative diffusion-generating preferences on all items to alleviate data sparsity.
- Our PDRec achieves significant and consistent improvements on different datasets, base SR models, and tasks. Its detachable components are well-received in practice.

## 2 Related Work

**Sequential recommendation.** Sequential Recommendation (SR) is one of the representative methods for capturing users’ dynamic temporal-aware preference evolution patterns by modeling the sequential dependencies of their historical behaviors, thereby recommending the next item that the user may be interested in (Li et al. 2022). In recent years, Convolutional Neural Network (CNN) (Xu et al. 2019; Tang and Wang 2018), Recurrent Neural Network (RNN) (Li et al. 2017; Hidasi et al. 2016) and Transformer (Sun et al. 2019; Kang and McAuley 2018) are introduced into SR to capture users’ preference dependencies. GRU4Rec (Hidasi et al. 2016) employs the Gate Recurrent Unit (GRU) as the sequential encoder to learn users’ long-term dependencies. SASRec (Kang and McAuley 2018), one of the most widely used methods in SR, introduces Transformers for historical behavior interaction modeling. CL4SRec (Xie et al. 2022) is a strong SR models that proposes three sequence-based augmentations to construct contrastive learning (CL) tasks in SR. Nevertheless, existing methods primarily focus on modeling users’ long- and short-term behaviors with various neural architectures, disregarding the potential impact of the time-interval-sensitive knowledge and the recommender’s generalization capability across the entire corpus on the modeling of user preferences.

**Diffusion models in recommendation.** As a prominent deep generative method, Diffusion Models (DM) are inspired by non-equilibrium statistical physics and have demonstrated exceptional performance in Super Resolution (Ho et al. 2022; Shi et al. 2022), Semantic Segmentation (Brempong et al. 2022), and Time Series Imputation (Tashiro et al. 2021). Despite this, the relevant studies in the field of recommendation are marked by a notable scarcity. CODIGEM (Walker et al. 2022) leverages DM to generate robust collaborative signals and latent representations by modeling intricate and non-linear patterns. DiffRec (Wang et al. 2023) reduces the added noises into the generative process to retain globally analogous yet personalized collaborative information in a denoising manner. DiffuRec (Li, Sun, and Li 2023) and DiffRec\* (Du et al. 2023) corrupts the item representations into the Gaussian distribution and reverses them based on the historical behaviors to employ uncertainty injection in item representation construction.

However, the former two DM works exhibit an excessive reliance on the top-ranked data derived from diffusion, not only resulting in computing consumption and homogeneous recommendation results but also disregarding the comprehensive user historical behaviors. While the latter share the same modeling pipeline with SR methods, thus can serve as the base SR model within PDRec. The proposed PDRec differs from these works: (a) instead of directly training a diffusion model, we smartly leverage a pre-trained DM model to diminish the time complexity in model training. (b) We achieve the dual enhancement of recommendation diversity and preference modeling through denoising, knowledge distillation and negative sampling on sequence encoders with the diffusion-based preference. (c) PDRec is model- and task-agnostic, enabling its application across different sequence encoders and recommendation scenarios.

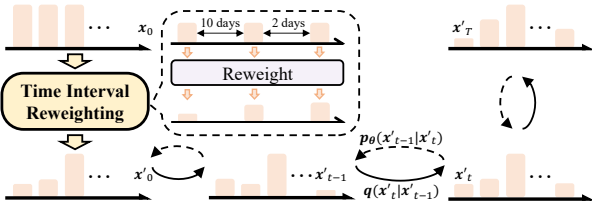


Figure 2: The illustration of our enhanced time-interval diffusion recommendation model (TI-DiffRec).

### 3 Time-Interval Diffusion Model

In this section, we present the Time-Interval Diffusion Recommendation model (TI-DiffRec). Following the classical DM methods (Ho, Jain, and Abbeel 2020; Nichol and Dhariwal 2021), the pioneering methods (Walker et al. 2022; Wang et al. 2023) typically leverage the original interaction matrix for diffusion, which is challenging to apply in SR. Despite incorporating the temporal order of user interactions, T-DiffRec (Wang et al. 2023) overlooks the time interval between consecutive behaviors, potentially leading to the issue of preference drift. As illustrated in Fig.2, we introduce an additional process of time interval reweighting alongside the diffusion and reverse process to tackle this challenge.

- **Time Interval Reweighting.** To incorporate the time interval information into diffusion, we first generate the time interval-aware input  $x'_0$  with the behavior sequence  $S_u = \{i_u^1, i_u^2, \dots, i_u^p\}$  and the corresponding timestamps  $T_u = \{t_u^1, t_u^2, \dots, t_u^p\}$  of user  $u$ . Subsequently, we compute the time-interval weight  $w_u^j = w_{\min} + \frac{t_u^j - t_u^1}{t_u^j - t_u^1} (w_{\max} - w_{\min})$  of each behavior  $i_u^j$ , where  $w_{\min}$  and  $w_{\max}$  denote the predefined lower and upper bounds. Thus we define  $x'_0 = [x_1, x_2, \dots, x_{|I|}]$  as the initial state for diffusion, where  $x_{i_u^j} = w_u^k$  or 0 indicates whether  $u$  has interacted with  $i_u^j$  or not ( $k$  denotes the index corresponding to  $i_u^j$  within  $S_u$ ).
- **Diffusion Process.** Generally, the diffusion process gradually injects uncertainty noise into the original data until the fully disordered state. The significant difference between DM and other latent variable models is that the transition kernel  $q(x'_t | x'_{t-1})$  used in DM obtains latent variables  $x'_t$  in a Markov chain process. Specifically, we employ the Gaussian perturbation as the transition kernel  $q(x'_t | x'_{t-1}) := \mathcal{N}(x'_t; \sqrt{1 - \beta_t} x'_{t-1}, \beta_t \mathbf{I})$ , where the variances  $\beta_t \in (0, 1)$  controls the Gaussian noise scales added at the step  $t$ . Note that the typical DM methods (Ho, Jain, and Abbeel 2020) fix the variances above to constants, demonstrating the notable property of DM that  $q$  has no learnable parameters. Thus, we can directly generate  $q(x'_t | x'_0) := \mathcal{N}(x'_t; \sqrt{\bar{\alpha}_t} x'_0, (1 - \bar{\alpha}_t) \mathbf{I})$  with the notation of  $\alpha_t := 1 - \beta_t$ ,  $\bar{\alpha}_t := \prod_{t'=1}^t \alpha_{t'}$  and  $t \in [1, 2, \dots, T]$ . If  $T \rightarrow +\infty$ ,  $x'_T$  asymptotically converges to the standard Gaussian distribution. That is, given  $x'_0$ , we can easily obtain  $x'_t = \sqrt{\bar{\alpha}_t} x'_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$  by sampling the Gaussian vector  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  via the reparameterization (Kingma and Welling 2013).

- **Reverse Process.** The reverse process aims to recover

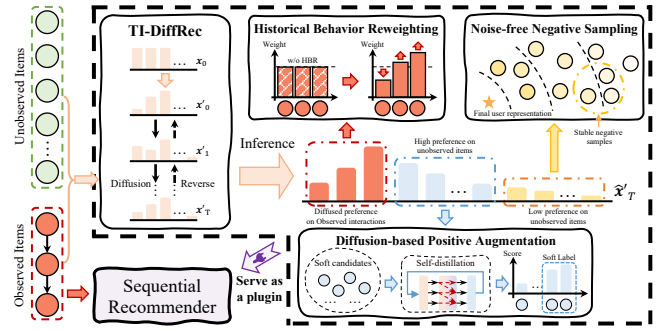


Figure 3: The overall structure of the proposed PDRec.

user’s interactions step by step from the standard Gaussian distribution through the denoising transition. Precisely, given the Gaussian distribution vector  $x'_T$ , we gradually remove the noise and recover the original interactions with the learnable transition kernel  $p_\theta(x'_{t-1} | x'_t)$  in the reverse direction. The reverse transition phase can be defined as:

$$p_\theta(x'_{t-1} | x'_t) := \mathcal{N}(x'_{t-1}; \mu_\theta(x'_t, t), \Sigma_\theta(x'_t, t)) \quad (1)$$

where  $\mu_\theta(x'_t, t)$  and  $\Sigma_\theta(x'_t, t)$  are parameterized by Deep Neural Networks (DNN) and  $\theta$  denotes model parameters. We can model complex interaction generation procedures for recommendation by such an iterative reverse process.

## 4 Plug-in Diffusion Model

### 4.1 Task Formulation and Overall Framework

SR aims to improve the next-item recommendation performance via the users’ historical sequential behaviors. To this end, given the *behavior sequence*  $S_u = \{i_u^1, i_u^2, \dots, i_u^p\}$  of user  $u \in \mathcal{U}$ , where  $i_j \in \mathcal{I}$  is the  $j$ -th behavior of  $u$  and  $p$  denote the historical behavior length, PDRec tries to recommend the target item  $i_u^{p+1}$  that will be interacted by this user.

In this section, we describe the proposed model-agnostic Plug-in Diffusion Model for Recommendation (PDRec) framework, which leverages the diffusion model as a flexible plugin to accurately model the dynamic user preferences in SR. The overall structure of PDRec is illustrated in Fig.3. Specifically, PDRec first explicitly regards the user behavior timestamps in diffusion to capture the dynamics of the actual sequential patterns and generate the user’s preferences on all items. Next, PDRec proposes a Historical Behavior Reweighting (HBR) strategy to identify specific indispensable supervised signals with the diffusion-based preferences on observed items. Additionally, PDRec designs a Diffusion-based Positive Augmentation (DPA) method to alleviate the data sparsity problem, which conducts self-distillation to incorporate the probable interactions from the unobserved items as the augmented soft samples into the training process in a dynamic manner. Finally, PDRec employs Noise-free Negative Sampling (NNS) to select stable negative samples, with the aim to mitigate the potential false negative problem. It is noteworthy that PDRec is task-agnostic, allowing such a framework can be easily migrated to cross-domain sequential recommendation (CDSR) tasks, and the related analysis is illustrated in Sec.5.5.

## 4.2 Historical Behavior Reweighting

The basic task of sequential preference modeling is to accurately and comprehensively leverage the historical behavior sequences. It is intuitive that different items within the user’s sequential sequence should hold varying degrees of importance for the next-item prediction. Therefore, the critical aspect of historical preference modeling lies in the adaptive and fine-grained differentiation of all observed items. This involves integrating the distinct importance of various items into the training phase. To this end, we propose a Historical Behavior Reweighting (HBR) strategy to reweight the supervised signals in training for behavior sequence denoising.

Specifically, given the pre-trained TI-DiffRec model and the complete interaction state  $\mathbf{x}_0$  of user  $u$  in the training set, we first obtain the time-interval-aware state  $\mathbf{x}'_0$ . Following the inference process in DiffRec (Wang et al. 2023), we regard the  $\mathbf{x}'_0$  which is naturally noisy and retaining personalized information as the noised state  $\hat{\mathbf{x}}'_T$ . Then we employ the reverse denoising by  $\hat{\mathbf{x}}'_T \rightarrow \dots \rightarrow \hat{\mathbf{x}}'_1 \rightarrow \hat{\mathbf{x}}'_0$  to generate the diffusion-based preferences  $\hat{\mathbf{x}}'_0 \in \mathbb{R}^{|\mathcal{I}|}$  of  $u$  on all items  $\mathcal{I}$ . To reweight the supervised signals, we exclusively focus on the observed preferences  $\mathbf{o}_u \in \mathbb{R}^{|\mathcal{I}_u^+|}$  and their corresponding ranking  $\mathbf{r}_u \in \mathbb{R}^{|\mathcal{I}_u^+|}$  from  $\hat{\mathbf{x}}'_0$  in HBR. The reweight vector  $\mathbf{w}_u$  to the supervised signals of user  $u$  is formulated as:

$$\hat{\mathbf{w}}_u = (1-\omega_r) \cdot \omega_s \cdot \frac{\mathbf{o}_u - \min_{\mathbf{o}_u}}{\max_{\mathbf{o}_u} - \min_{\mathbf{o}_u}} + \omega_r \cdot \frac{1 + \max_{\mathbf{r}_u} - \mathbf{r}_u}{\max_{\mathbf{r}_u}} \quad (2)$$

where  $\omega_s = \text{len}(S_u) / \text{sum}(\frac{\mathbf{o}_u - \min_{\mathbf{o}_u}}{\max_{\mathbf{o}_u} - \min_{\mathbf{o}_u}})$ ,  $\omega_r$  and  $(1-\omega_r)$  denote the ranking weight of observed ranking and preferences respectively. This mixture ensures that the recommender’s optimization direction remains reasonably aligned with the observed prior to the reweighting process. Finally, PDRec generates the final reweight vector  $\mathbf{w}_u = \omega_f \cdot \min(\max(c_w, \min(\hat{\mathbf{w}}_u)), \max(\hat{\mathbf{w}}_u))$  by truncating and rescaling the reweight vector via the truncate value  $c_w$  and the rescale weight  $\omega_f$  to prevent certain signals dominate the optimization process. Note that PDRec only employs the inference process before model training without introducing excessive computational cost. With HBR, we can not only directly focus on the time-interval-aware preferences related to the user’s behaviors, but also leverage DM’s informative uncertainty to denoise the dispensable items and highlight the indispensable actions in the user’s behavior sequence.

## 4.3 Diffusion-based Positive Augmentation

Inspired by the promising performance of TI-DiffRec in SR and its inherent strong generalization characteristics, PDRec assumes that the user’s diffusion-based preferences  $\hat{\mathbf{x}}'_0$  on unobserved items encompass specific samples that user  $u$  is potentially interested in but have never seen before. This intuitive observation exhibits heightened prominence within the top-ranked range of the user’s diffusion-based preference. To transfer the generalized knowledge from the pre-trained TI-DiffRec to the sequential recommender, PDRec intelligently designs a Diffusion-based Positive Augmentation (DPA) method to distil the essential diffusion-based information by regarding the unobserved items with high preferences as the potential positive samples during training.

To do it, PDRec first takes top-ranked  $m$  items to form the potential soft samples  $t_u$  based on the diffusion-based unobserved preferences  $\mathbf{u}_u = \hat{\mathbf{x}}'_0 \mathbf{o}_u$ , where  $\hat{\mathbf{x}}'_0$  and  $\mathbf{o}_u$  denote the diffusion-based preferences of the corpus and supervised signals. Following the assumption that “the last behavior in a user’s behavioral sequence reflects his/her overall interests”, PDRec calculate the matching score  $m_u = [(\mathbf{h}_u)^\top \mathbf{t}_1, (\mathbf{h}_u)^\top \mathbf{t}_2, \dots, (\mathbf{h}_u)^\top \mathbf{t}_m]$  between the user’s last behavior representation  $\mathbf{h}_u$  obtained by SASRec (Kang and McAuley 2018) and the item embedding matrix  $\mathbf{T}_u = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_m]$  of the potential soft samples  $t_u$ . After re-ranking the matching score  $m_u$ , PDRec extract the top-ranked  $n$  items to obtain the soft positive augmentations  $s_u$ . The optimization approach for  $s_u$  will be defined in Sec. 4.5.

## 4.4 Noise-free Negative Sampling

Existing recommendation algorithms generally require both positive and negative examples to model users’ personalized preferences. They expect explicit interactions from the dataset ideally. However, explicit feedback is not always available in real-world scenarios, the ubiquitous users’ implicit interactions may not necessarily reflect their real interests. Conventional recommenders typically employ uniform probability for negative sampling, which fail to consider the dynamic shifts in user preferences, potentially leading to the false negative problem to some extent. Inspired by the exploration of negative sampling strategies in recommendation (Shi et al. 2023; Ma et al. 2023b), PDRec introduces the Noise-free Negative Sampling (NNS) strategy to prioritize the unobserved samples with low-scored diffusion-based preference and select safe negative samples to direct HBR and DPA in the stable optimization direction.

In contrast to the DPA utilizing the unobserved items with high preferences as the soft positive augmentations, NNS creatively regard these items with low preferences as the additional negative samples in training. Precisely, given the diffusion-based unobserved preferences  $\mathbf{u}_u$ , PDRec sorts them, selects low-scored items from the unobserved corpus  $\mathcal{I}_u^- = \mathcal{I} \setminus \mathcal{I}_u^+$ , and assigns higher sampling probabilities to these items. The sampling probability of NNS is defined as:

$$P^{NNS}(j | \mathcal{I}_u^-) = \begin{cases} \frac{1}{(1-\omega_m)l_u}, & j \in K_u[\omega_m l_u : l_u] \\ 0, & j \in \text{others} \end{cases} \quad (3)$$

where  $K_u$  is the re-ranked item list of  $\mathcal{I}_u^-$  obtained by sorting the diffused unobserved preferences  $\mathbf{u}_u$ ,  $l_u = |K_u|$  denotes the number of unobserved items and  $\omega_m$  denotes the initial proportion of the negative sampling. Note that the bigger the  $\omega_m$  is, the more stable the samples will be drawn.

## 4.5 Optimization Objectives

We calculate the predicted probability  $\hat{y} = (\mathbf{h}_u)^\top \mathbf{v}_{q+1}$  with the sequence representation  $\mathbf{h}_u$  of user  $u$  and the item embedding  $\mathbf{v}_{q+1}$ . Then we formulate the Binary Cross-Entropy loss  $\mathcal{L}_R$  and the self-distillation loss  $\mathcal{L}_D$  in DPA as follows:

$$\mathcal{L}_R = - \sum_{(u,i) \in R} [w_u \cdot y_{u,i} \log \hat{y}_{u,i} + (1 - y_{u,i}) \log(1 - \hat{y}_{u,i})] \quad (4)$$

$$\mathcal{L}_D = - \sum_{(u,i) \in R^+} [y_{u,i} \log \hat{y}_{u,i}] \quad (5)$$

Datasets	Metrics	T-DiffRec	TI-DiffRec	GRU4Rec	+PDR	Improv.	SASRec	+PDR	Improv.	CL4SRec	+PDR	Improv.
Toy	N@1	0.1033	0.1058	0.0878	<b>0.0899</b>	2.39%	0.1095	<b>0.1247</b>	13.88%	0.1125	<b>0.1254</b>	11.47%
	N@5	0.1564	0.1618	0.1515	<b>0.1617</b>	6.73%	0.1779	<b>0.2023</b>	13.72%	0.1802	<b>0.2041</b>	13.26%
	N@10	0.1758	0.1823	0.1755	<b>0.1879</b>	7.07%	0.2020	<b>0.2286</b>	13.17%	0.2046	<b>0.2305</b>	12.66%
	HR@5	0.2055	0.2151	0.2128	<b>0.2300</b>	8.08%	0.2423	<b>0.2752</b>	13.58%	0.2438	<b>0.2776</b>	13.86%
	HR@10	0.2657	0.2787	0.2874	<b>0.3112</b>	8.28%	0.3169	<b>0.3568</b>	12.59%	0.3195	<b>0.3595</b>	12.52%
	AUC	0.5911	0.5968	0.5670	<b>0.5909</b>	4.22%	0.5771	<b>0.6060</b>	5.01%	0.5805	<b>0.6068</b>	4.53%
Game	N@1	0.1611	0.1746	0.1667	<b>0.1808</b>	8.46%	0.2111	<b>0.2191</b>	3.79%	0.2106	<b>0.2180</b>	3.51%
	N@5	0.2567	0.2723	0.2818	<b>0.2996</b>	6.32%	0.3310	<b>0.3382</b>	2.18%	0.3294	<b>0.3368</b>	2.25%
	N@10	0.2895	0.3040	0.3199	<b>0.3380</b>	5.66%	0.3682	<b>0.3753</b>	1.93%	0.3682	<b>0.3750</b>	1.85%
	HR@5	0.3451	0.3618	0.3893	<b>0.4091</b>	5.09%	0.4409	<b>0.4475</b>	1.50%	0.4385	<b>0.4456</b>	1.62%
	HR@10	0.4469	0.4600	0.5071	<b>0.5282</b>	4.16%	0.5559	<b>0.5626</b>	1.21%	0.5584	<b>0.5638</b>	0.97%
	AUC	0.7217	0.7234	0.7601	<b>0.7786</b>	2.43%	0.7865	<b>0.7908</b>	0.61%	0.7857	<b>0.7905</b>	0.61%
Book	N@1	0.3194	0.3275	0.3072	<b>0.3359</b>	9.34%	0.3594	<b>0.3656</b>	1.73%	0.3554	<b>0.3621</b>	1.89%
	N@5	0.4398	0.4491	0.4433	<b>0.4757</b>	7.31%	0.4948	<b>0.5063</b>	2.32%	0.4942	<b>0.5047</b>	2.12%
	N@10	0.4671	0.4776	0.4765	<b>0.5091</b>	6.84%	0.5272	<b>0.5393</b>	2.30%	0.5276	<b>0.5376</b>	1.90%
	HR@5	0.5459	0.5557	0.5643	<b>0.6004</b>	6.40%	0.6148	<b>0.6306</b>	2.57%	0.6166	<b>0.6304</b>	2.24%
	HR@10	0.6300	0.6435	0.6667	<b>0.7033</b>	5.49%	0.7150	<b>0.7323</b>	2.42%	0.7197	<b>0.7317</b>	1.67%
	AUC	0.8160	0.8202	0.8541	<b>0.8728</b>	2.19%	0.8790	<b>0.8898</b>	1.23%	0.8820	<b>0.8895</b>	0.85%
Music	N@1	0.3401	0.3494	0.3299	<b>0.3540</b>	7.31%	0.3753	<b>0.3826</b>	1.95%	0.3689	<b>0.3755</b>	1.79%
	N@5	0.4709	0.4773	0.4725	<b>0.5000</b>	5.82%	0.5170	<b>0.5283</b>	2.19%	0.5096	<b>0.5211</b>	2.26%
	N@10	0.4987	0.5049	0.5069	<b>0.5348</b>	5.50%	0.5503	<b>0.5620</b>	2.13%	0.5435	<b>0.5558</b>	2.26%
	HR@5	0.5852	0.5886	0.5987	<b>0.6287</b>	5.01%	0.6421	<b>0.6573</b>	2.37%	0.6353	<b>0.6504</b>	2.38%
	HR@10	0.6706	0.6738	0.7048	<b>0.7361</b>	4.44%	0.7447	<b>0.7612</b>	2.22%	0.7400	<b>0.7573</b>	2.34%
	AUC	0.8329	0.8318	0.8768	<b>0.8908</b>	1.60%	0.8962	<b>0.9040</b>	0.87%	0.8939	<b>0.9026</b>	0.97%

Table 1: Results between backbones and PDR on four datasets. All improvements are significant ( $p < 0.05$  with paired t-tests).

Datasets	Toy	Game	Book	Music
Users	7,996	7,996	12,170	12,170
Items	37,868	11,735	33,697	30,707
Records	114,487	82,871	514,015	558,352
Density	0.0378%	0.0883%	0.1253%	0.1494%

Table 2: Statistics of four SR datasets.

where  $R$  denotes the training set which contains the supervised signals, the random negative samples and the safe negative items sampled within NNS,  $R^+$  denotes the soft positive augmentations  $s_u$  in DPA,  $w_u$  is the final reweight vector in HBR,  $y_{u,i} = 1/0$  denote the positive and the sampled negative pairs respectively, and  $\hat{y}_{u,i}$  denotes the predicted probability of  $(u, i)$ . To optimize in conjunction with the self-distillation augmentation, the objective function  $\mathcal{L}$  is a linear combination of  $\mathcal{L}_R$  and  $\mathcal{L}_D$  with the loss weight  $\omega_d$  of  $\mathcal{L}_D$ :

$$\mathcal{L} = \mathcal{L}_R + \omega_d \mathcal{L}_D. \quad (6)$$

## 5 Experiments

In this section, we conduct extensive experiments and analyses for the following research questions: (RQ1): How does PDR perform against the state-of-the-art SR baselines (see Sec.5.2)? (RQ2): How do different components of PDR benefit its performance (see Sec.5.3)? (RQ3): Is PDR still effective with other base SR models (see Sec.5.4)? (RQ4): Could PDR be further adopted to other tasks such as cross-domain sequential recommendation? (see Sec.5.5)?

### 5.1 Experimental Settings

**Dataset.** We conduct extensive experiments on four real-world datasets. We select ‘‘Toys and Games’’ and ‘‘Video Games’’ to form the *Toy* and *Game* dataset from Amazon (Lin et al. 2022). From Douban, we pick ‘‘Books’’ and ‘‘Musics’’ to form the *Book* and *Music* dataset (Wu et al. 2023).

**Baselines.** We implement PDR on three representative SR models: **GRU4Rec** (Hidasi et al. 2016), **SASRec** (Kang and McAuley 2018) and **CL4SRec** (Xie et al. 2022), and compare it with **T-DiffRec** (Wang et al. 2023) to validate its effectiveness and universality. Note that T-DiffRec (Wang et al. 2023) is one of the SOTA DM-based recommenders that captures the temporal patterns in user interactions.

**Parameter settings.** For fair comparisons, we set the learning rate and the maximum sequence length as  $5e^{-3}$  and 200. According to the natural distribution of behaviors, we set the  $\omega_m$  as 0.5 for the relatively sparse Amazon datasets and 0.8 for the denser Douban datasets. Similarly, we define the number of coarse-grained sorted items  $m$ , the number of fine-grained resorted items  $n$ , and the loss weight  $\omega_d$  of  $\mathcal{L}_{SD}$  as 50, 5 and 0.3 for Amazon. For Douban, these parameter are configured as 100, 1, and 0.01, respectively. Due to the variations in TI-DiffRec’s confidence range, PDR exhibits minor discrepancies in the parameters of HBR across diverse datasets. That is, the ranking weight  $\omega_r$ , the truncate value  $c_w$  and the rescale weight  $\omega_f$  are denoted as 0.1, 3 and 2 for Toy, 0.1, 5 and 4 for Game, 0.3, 3 and 4 for Book and 0.1, 5 and 2 for Music. Each experiment is conducted five times with random seeds, and we report the average results.



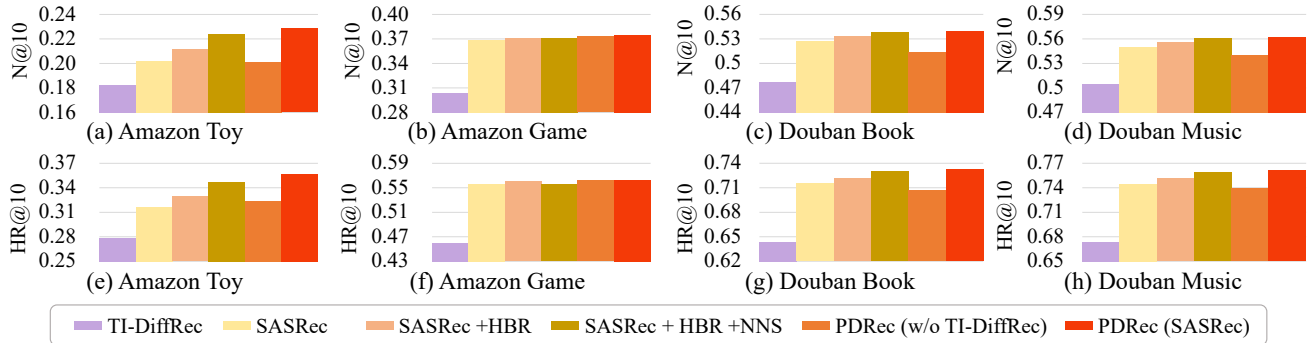


Figure 4: Results on ablation study of PDRec (SASRec) on four datasets. Generally, all components are effective.

## 5.2 Performance Comparison on SR (RQ1)

We conduct the experiments on four public datasets, adopting three typical evaluation metrics, including NDCG@k ( $N@k$ ), Hit Rate@k ( $HR@k$ ), and AUC with different  $k = 1, 5, 10$ . Following (Kang and McAuley 2018), we randomly sample 99 negative items for each positive instance in testing. Table 1 shows the overall performance comparison results, the best results of the same backbone are in boldface. It reveals the following observations:

(1) In general, PDRec significantly outperforms all baselines on four datasets, exhibiting the significance level  $p < 0.05$  and the average error range  $\leq 0.004$ . This indirectly confirms the significance of (a) the observed interactions denoising is able to guide the recommender toward an accurate and unbiased optimization direction; (b) the handling of positive and negative aspects of unobserved interactions effectively leverages the informative yet user-imperceptible knowledge from the diffusion model, expanding user interests while stabilizing the training process.

(2) Comparing the improvements across various datasets, we discover that PDRec benefits the relatively sparse Toy and Game datasets more. Meanwhile, PDRec can obtain promising performance even on denser datasets. Furthermore, we also observed that PDRec, implemented with diverse backbones, consistently exhibits significant improvements over its respective backbones. This may be attributed to the precise utilization of diffusion models, PDRec can assist in highlighting the actual long- and short-term sequential dependencies. As a task-agnostic framework, we further expand PDRec into the field of CDSR to employ the feasibility analyses on recommendation scenarios in Sec. 5.5.

(3) Simultaneously, we notice the significant improvement of the proposed TI-DiffRec relative to T-DiffRec (Wang et al. 2023), underscoring the necessity of time-interval knowledge in SR. Nevertheless, the performance of these DM-based algorithms remains inferior to the existing SOTA sequential recommendation algorithms. In conjunction with the notable improvement of PDRec over these SR methods, the effectiveness of the proposed PDRec is firmly established. It can smartly combine the sequential modeling capability of (the future advanced) SR models and the potent generalized ability of diffusion models on the corpus, thus precisely accomplishing sequential recommendation tasks.

## 5.3 Ablation Study (RQ2)

In this section, we conduct ablation studies to explore the effectiveness of different components in PDRec. Thus we compare PDRec (SASRec) with different ablation versions of PDRec to verify the benefits of TI-DiffRec, HBR, DPN and NNS, respectively. Note that PDRec (SASRec) equals SASRec+HBR+NNS+DPA. From Fig. 4 we observe that:

(1) With HBR, SASRec+HBR achieves consistent improvement over SASRec. This mainly stems from the fact that diffusion-based preferences generated by the powerful TI-DiffRec can effectively denoise the historical behaviors via reweighting. It enables the recommender to emphasize the indispensable supervised signals while disregarding noisy interactions, thereby enhancing training efficiency.

(2) Comparing SASRec+HBR+NNS to SASRec+HBR, we find that NNS yields performance gains across most datasets. It demonstrates that these “safe” negative items judged by previous diffusion models could aid in alleviating the inherent false negative problems in model training.

(3) PDRec further improves the performance of SASRec+HBR+NNS. DPA emphasizes the top-ranked preferences determined by the diffusion model for unobserved items, thereby inferring user’s more diverse potential preferences given by diffusion models. By double checking these high-quality positive augmentation candidates via self-distillation, DPA could bring in additional positive signals via a more flexible way to fight against data sparsity.

(4) PDRec achieves significant improvement compared to PDRec without TI-DiffRec (i.e., replacing TI-DiffRec with another SASRec). It highlights the necessity of employing diffusion models. Owing to the problem formulation, DM preserves the visibility into all items in the corpus. In conjunction with its powerful generalization ability, DM can offer informative knowledge relative to sequential models (i.e., SASRec), particularly for sparse user-item interaction matrices. Nevertheless, compared to the original DiffRec, PDRec is more effective and practical.

## 5.4 Universality Analysis of PDRec (RQ3)

PDRec is a model-agnostic framework. To verify this, we employ each ablation variant of PDRec over GRU4Rec (Hidasi et al. 2016) and CL4SRec (Xie et al. 2022) on Toy and Game datasets. Fig. 5 illustrates the results. We can find that:

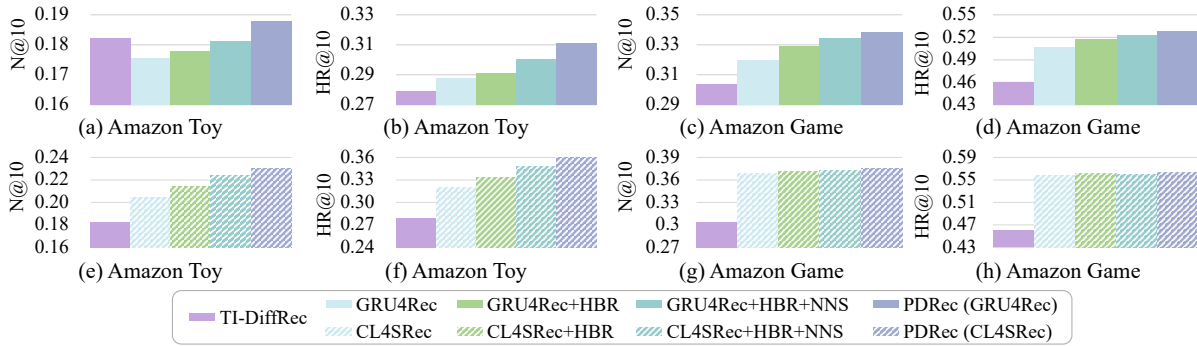


Figure 5: Results of PDRec on GRU4Rec/CL4SRec and their ablation versions on Toy and Game datasets.

Setting	Algorithms	N@1	N@5	N@10	N@20	N@50	HR@5	HR@10	HR@20	HR@50	AUC
Game ↓ Toy	<b>T-DiffRec (M)</b>	0.0981	0.1520	0.1727	0.1934	0.2375	0.2029	0.2673	0.3494	0.5780	0.5924
	<b>TI-DiffRec (M)</b>	0.1053	0.1598	0.1806	0.2008	0.2407	0.2111	0.2759	0.3562	0.5623	0.5932
	<b>SASRec (M)</b>	0.1267	0.2019	0.2261	0.2490	0.2785	0.2722	0.3472	0.4380	0.5873	0.5951
	<b>+HBR</b>	0.1283	0.2061	0.2311	0.2533	0.2835	0.2785	0.3558	0.4438	0.5972	0.6092
	<b>+HBR+NNS</b>	0.1264	0.2068	0.2323	0.2542	0.2844	0.2815	0.3606	0.4480	0.6013	<b>0.6123</b>
	<b>+HBR+NNS+DPA</b>	<b>0.1302</b>	<b>0.2093</b>	<b>0.2348</b>	<b>0.2574</b>	<b>0.2873</b>	<b>0.2826</b>	<b>0.3616</b>	<b>0.4515</b>	<b>0.6026</b>	0.6106
Toy ↓ Game	<b>T-DiffRec (M)</b>	0.1674	0.2643	0.2977	0.3247	0.3597	0.3548	0.4584	0.5655	0.7428	0.7232
	<b>TI-DiffRec (M)</b>	0.1709	0.2757	0.3096	0.3378	0.3721	0.3723	0.4773	0.5887	0.7622	0.7407
	<b>SASRec (M)</b>	0.2273	0.3532	0.3905	0.4190	0.4467	0.4674	0.5826	0.6955	0.8342	0.8007
	<b>+HBR</b>	0.2332	0.3597	0.3963	0.4250	0.4547	0.4741	0.5872	0.7006	0.8501	0.8145
	<b>+HBR+NNS</b>	0.2352	0.3601	0.3975	0.4257	0.4557	0.4733	0.5890	0.7002	0.8517	0.8138
	<b>+HBR+NNS+DPA</b>	<b>0.2363</b>	<b>0.3623</b>	<b>0.3992</b>	<b>0.4275</b>	<b>0.4572</b>	<b>0.4761</b>	<b>0.5904</b>	<b>0.7022</b>	<b>0.8520</b>	<b>0.8153</b>

Table 3: Ablation versions of PDRec on two CDSR datasets. All improvements are significant compared to baselines.

(1) PDRec achieves significant improvements over different base models (GRU4Rec and CL4SRec) across diverse datasets. This demonstrates the universality of PDRec on different sequential encoders. Furthermore, it indirectly underscores the potential of PDRec to leverage the possible advancements in SR in the future, thereby extending the lifespan of the proposed DM-utilization frameworks.

(2) Progressive improvements are discernible among distinct versions of PDRec, with PDRec outperforming all its variants. It demonstrates that the proposed components are effective and universal for different base sequential encoders and datasets, further reconfirming the universality of PDRec.

### 5.5 Results of Cross-domain SR (RQ4)

PDRec could also benefit positive transfer in CDSR. We follow typical CDSR settings (Ma et al. 2023c; Zheng et al. 2022) and employ PDRec with SASRec (M) (M indicates directly mixing both source and target domains’ behaviors in chronological order) on Toy→Game and Game→Toy settings. We also implement T-DiffRec (M) and TI-DiffRec (M) on the mixed (M) setting. From Table. 3, we have:

(1) PDRec outperforms all diffusion-based models in CDSR, which implies that PDRec could be used in other tasks such as cross-domain scenarios. Its HBR provides an intuitive but effective way to filter negative transfers in cross-domain recommendation (i.e., mixing all domains’ behaviors sequentially and conducting reweighting via diffu-

sion), which could be further explored in the future.

(2) PDRec outperforms all of its ablation versions on most CDSR settings, with each component contributing to incremental improvements. It reconfirms the effectiveness and universality of HBR, NNS, and DPA from diffusion model.

(3) It is impressive that PDRec exhibits notable improvements across various metrics compared to the original T-DiffRec/TI-DiffRec on the mixed behavior sequence (**up to 38.3%**). It reiterates our main contribution that takes full advantage of the outputs of diffusion model as a plugin in SR.

## 6 Conclusion

In this paper, we propose an effective and model-agnostic Plug-in Diffusion Model for Recommendation (PDRec) framework. Instead of focusing on the highest-score item, PDRec fully leverages the diffusion-based preferences on all items. PDRec employs a historical behavior reweighting method to identify the indispensable behaviors and conducts a knowledge extracting strategy from both the unobserved items via the diffusion-based positive augmentation, and noise-free negative sampling. The extensive experiments and analyses on four datasets, three base models and two recommendation tasks demonstrate the effectiveness and universality of PDRec. In the future, we will continue to explore the tailored hard negative sampling strategies in PDRec and attempt to adapt PDRec as a flexible and detachable plugin in diverse recommendation scenarios.

## 7 Acknowledgments

This work is supported in part by the TaiShan Scholars Program (Grant no. tsqn202211289), the National Natural Science Foundation of China (Grant no. 62006141), the Excellent Youth Scholars Program of Shandong Province (Grant no. 2022HWYQ-048) and the Oversea Innovation Team Project of the "20 Regulations for New Universities" funding program of Jinan (Grant no. 2021GXRC073). ChatGPT and Grammarly were utilized to improve grammar and correct spelling.

## References

- Brempong, E. A.; Kornblith, S.; Chen, T.; Parmar, N.; Minderer, M.; and Norouzi, M. 2022. Denoising pretraining for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.
- Chen, G.; Zhang, X.; Su, Y.; Lai, Y.; Xiang, J.; Zhang, J.; and Zheng, Y. 2023a. Win-Win: A Privacy-Preserving Federated Framework for Dual-Target Cross-Domain Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
- Chen, H.; He, J.; Xu, W.; Feng, T.; Liu, M.; Song, T.; Yao, R.; and Qiao, Y. 2023b. Enhanced Multi-Relationships Integration Graph Convolutional Network for Inferring Substitutable and Complementary Items. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
- Chen, Y.; Liu, Z.; Li, J.; McAuley, J.; and Xiong, C. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference (WWW)*.
- Du, H.; Yuan, H.; Huang, Z.; Zhao, P.; and Zhou, X. 2023. Sequential Recommendation with Diffusion Models.
- Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of International Conference on Learning Representations (ICLR)*.
- Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Proceedings of Advances in neural information processing systems ((NeurIPS))*.
- Ho, J.; Saharia, C.; Chan, W.; Fleet, D. J.; Norouzi, M.; and Salimans, T. 2022. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research (JMLR)*.
- Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *Proceedings of International Conference on Data Mining (ICDM)*.
- Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; and Ma, J. 2017. Neural attentive session-based recommendation. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*.
- Li, M.; Zhao, X.; Lyu, C.; Zhao, M.; Wu, R.; and Guo, R. 2022. MLP4Rec: A Pure MLP Architecture for Sequential Recommendations.
- Li, Z.; Sun, A.; and Li, C. 2023. DiffuRec: A Diffusion Model for Sequential Recommendation. *arXiv preprint arXiv:2304.00686*.
- Lin, G.; Gao, C.; Li, Y.; Zheng, Y.; Li, Z.; Jin, D.; and Li, Y. 2022. Dual Contrastive Network for Sequential Recommendation with User and Item-Centric Perspectives. *arXiv preprint arXiv:2209.08446*.
- Lopez Alcaraz, J. M.; and Strodthoff, N. 2023. Diffusion-based time series imputation and forecasting with structured state space models. *Transactions on machine learning research (TMLR)*.
- Ma, H.; Li, X.; Meng, L.; and Meng, X. 2021. Comparative study of adversarial training methods for cold-start recommendation. In *Proceedings of ADVM*.
- Ma, H.; Qi, Z.; Dong, X.; Li, X.; Zheng, Y.; and Meng, X. M. L. 2023a. Cross-Modal Content Inference and Feature Enrichment for Cold-Start Recommendation. *Proceedings of IJCNN*.
- Ma, H.; Xie, R.; Meng, L.; Chen, X.; Zhang, X.; Lin, L.; and Zhou, J. 2023b. Exploring False Hard Negative Sample in Cross-Domain Recommendation. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*.
- Ma, H.; Xie, R.; Meng, L.; Chen, X.; Zhang, X.; Lin, L.; and Zhou, J. 2023c. Triple Sequence Learning for Cross-domain Recommendation. *arXiv preprint arXiv:2304.05027*.
- Meng, L.; Feng, F.; He, X.; Gao, X.; and Chua, T.-S. 2020. Heterogeneous fusion of semantic and collaborative information for visually-aware food recommendation. In *Proceedings of MM*.
- Moon, J.; Jeong, Y.; Chae, D.-K.; Choi, J.; Shim, H.; and Lee, J. 2023. CoMix: Collaborative filtering with mixup for implicit datasets. *Information Sciences*.
- Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *Proceedings of International Conference on Machine Learning (ICML)*. PMLR.
- Shi, W.; Chen, J.; Feng, F.; Zhang, J.; Wu, J.; Gao, C.; and He, X. 2023. On the Theories Behind Hard Negative Sampling for Recommendation. In *Proceedings of the ACM Web Conference (WWW)*.
- Shi, Y.; De Bortoli, V.; Deligiannidis, G.; and Doucet, A. 2022. Conditional simulation using diffusion Schrödinger bridges. In *Proceedings of Uncertainty in Artificial Intelligence (UAI)*.
- Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*.
- Tang, J.; and Wang, K. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of ACM International Conference on Web Search and Data Mining (WSDM)*.
- Tashiro, Y.; Song, J.; Song, Y.; and Ermon, S. 2021. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*.



Walker, J.; Zhong, T.; Zhang, F.; Gao, Q.; and Zhou, F. 2022. Recommendation via collaborative diffusion generative model. In *Proceedings of International Conference on Knowledge Science, Engineering and Management*, 593–605. Springer.

Wang, W.; Xu, Y.; Feng, F.; Lin, X.; He, X.; and Chua, T.-S. 2023. Diffusion Recommender Model. *Proceedings of International Conference on Research on Development in Information Retrieval (SIGIR)*.

Wu, B.; He, X.; Wu, L.; Zhang, X.; and Ye, Y. 2023. Graph-augmented co-attention model for socio-sequential recommendation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems (SMC)*.

Xia, L.; Huang, C.; Xu, Y.; Dai, P.; Zhang, X.; Yang, H.; Pei, J.; and Bo, L. 2021. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.; Ding, B.; and Cui, B. 2022. Contrastive learning for sequential recommendation. In *Proceedings of IEEE International Conference on Data Engineering (ICDE)*.

Xu, C.; Zhao, P.; Liu, Y.; Xu, J.; S. Sheng, V. S. S.; Cui, Z.; Zhou, X.; and Xiong, H. 2019. Recurrent convolutional neural network for sequential recommendation. In *Proceedings of International World Wide Web Conferences (WWW)*.

Zhang, M.; Wu, S.; Yu, X.; Liu, Q.; and Wang, L. 2022. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.

Zheng, X.; Su, J.; Liu, W.; and Chen, C. 2022. DDGHM: Dual Dynamic Graph with Hybrid Metric Training for Cross-Domain Sequential Recommendation. In *Proceedings of ACM International Conference on Multimedia (ACM MM)*.