Enhancing Diffusion Model with Auxiliary Information Mining-Exploration and Efficient Sampling Mechanism for Sequential Recommendation

Te Song[1], Lianyong Qi[1*], Weiming Liu[2], Fan Wang[2], Xiaolong Xu[3], Xuyun Zhang[4], Amin Beheshti[4], Xiaokang Zhou[5,6], Wanchun Dou[7]

[1]China University of Petroleum (East China), China
[2]Zhejiang University, China
[3]Nanjing University of Information Science and Technology, China
[4]Macquarie University, Australia
[5]Faculty of Business Data Science, Kansai University, Japan
[6]RIKEN Center for Advanced Intelligence Project, Japan
[7]Nanjing University, China
z23070102@s.upc.edu.cn, lianyongqi@upc.edu.cn, {21831010, fanwang97}@zju.edu.cn, xlxu@nuist.edu.cn
{xuyun.zhang, amin.beheshti}@mq.edu.au, zhou@kansai-u.ac.jp, douwc@nju.edu.cn

Abstract

Sequential recommendation aims to capture the temporal dependencies of items in a user's historical interactions and make recommendations based on this. Previous generative methods addressed the issue of data not directly reflecting user preference uncertainty by modeling the distribution of latent item representations. Diffusion model (DM)-based methods have achieved significant success due to their high-quality generation and stable training. However, they lack satisfactory user sequence representations to guide the generation process, impacting recommendation performance. Moreover, these methods overlook the drawback of slow inference speed, severely limiting their practical value. To obtain effective generative guidance signals and accelerate the recommendation process, we propose DAE4Rec. In this approach, a Graph Auto-Encoder (GAE) is used to obtain interpretable item node representations, revealing global transitions of items that previous methods struggled to uncover. Then, we use it to construct a generative guidance signal with lower coupling and variance for the diffusion model. Additionally, by employing a non-Markov chain derived from the forward diffusion process, it is the first to implement a 'skip-step' reverse process in diffusion model-based methods. And a creatively designed compensator is used to bridge the performance gap caused by 'skip-step'. Extensive experiments on three real-world datasets demonstrate that DAE4Rec outperforms other state-of-the-art generative sequential recommenders.

1 Introduction

Recommender systems (RSs) personalize user experiences by filtering vast online content in e-commerce, media streaming, and social networks (He et al. 2017; Wang et al. 2022a; Liu et al. 2022a, 2023b, 2024b). Sequential recommendations (SRs), an extension of RSs, capture temporal dependencies in user behavior to predict the next preferred item, enhancing recommendation effectiveness (Chen et al. 2018; Wang et al. 2019). Due to their practical value, SRs have been widely researched (Wu et al. 2017; Yang et al. 2023).

Many SR methods (Kang and McAuley 2018; Sun et al. 2019; Liu et al. 2023a) assume an accurate mapping between a user's historical interactions and the target item. However, this mapping is often challenging to capture, as user preferences may not always be reflected in their past interactions (Wang et al. 2022b), exhibiting uncertainty (e.g., occasional interest in novel items) (Li, Sun, and Li 2023).

Generative models address this issue by estimating and generating data based on probability distributions (Wang et al. 2024a; Liu et al. 2021, 2024a). Specifically, Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) and Variational Autoencoders (VAEs) (Kingma and Welling 2013; Liu et al. 2022b) model latent variable distributions for sequences to capture uncertainty (Li, Sun, and Li 2023). However, GANs struggle with training instability (Wang et al. 2024b), and VAEs face posterior collapse (Zhao, Song, and Ermon 2019), limiting recommendation performance.

Fortunately, emerging Diffusion Models (DMs) (Sohl-Dickstein et al. 2015) offer high-quality samples generation and stable training (Song, Meng, and Ermon 2020), achieving success in image synthesis (Dhariwal and Nichol 2021; Nichol et al. 2021; Rombach et al. 2022; Song et al. 2020), text generation (Li et al. 2022), and human motion generation (Tevet et al. 2022). Building on this, generating data according to user intentions (e.g., an image of a cat) has gained significant attention (Rombach et al. 2022; Saharia et al. 2022). The conditional guidance mechanism injects additional information (e.g., class labels or text sequences) into the denoiser during diffusion, with this guidance referred to as the *condition*. This concept aligns well with SR by using user interaction sequences as the condition (Yang et al. 2024), providing meaningful sequence information for generating the target item and ensuring recommendations align more closely with user preferences.

*Corresponding Author.

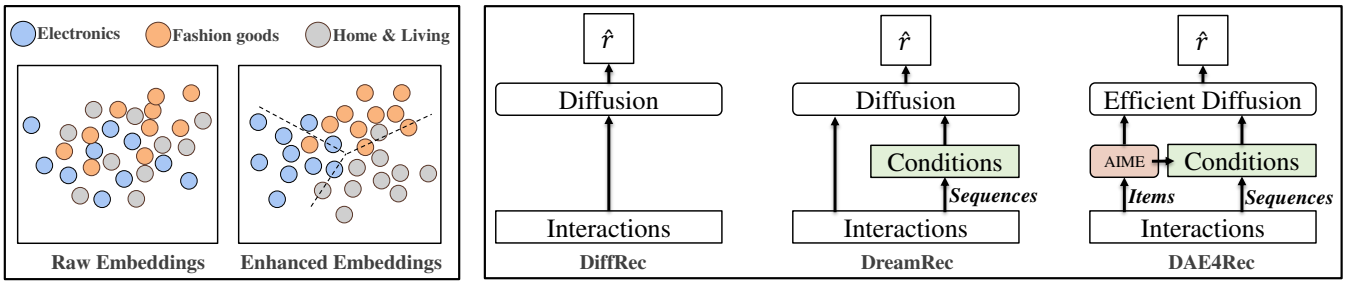


Figure 1: An illustration of unsatisfactory raw item embeddings and our enhanced item embeddings (left sub-figure); framework comparison of representational methods (right sub-figure). In the left sub-figure, differently colored circles represent different types of item embeddings, the previous method's insufficient consideration of global item dependence led to highly difference lacking. In contrast, our approach DAE4Rec enhances embeddings by designed AIME module as shown in the right sub-figure.

Therefore, this paper focuses on sequential recommendation based on conditional diffusion models. While these methods have achieved some success in recommendation (Li, Sun, and Li 2023; Wang et al. 2024b; Yang et al. 2024), two unresolved challenges remain in previous work:

(1) Unsatisfactory Condition Representation. Effectively encoding items and representing conditional signals remains underexplored, impacting generation quality and recommendation personalization. Early methods like DiffRec (Wang et al. 2023), based on unconditional DM, reshaped collaborative filtering by generating interaction matrices but failed to model implicit user preferences, limiting personalization. Subsequent methods, such as DreamRec (Yang et al. 2024), use sequence learning models to capture item dependencies and construct sequence representations to guide diffusion sampling. However, they mostly rely on simple discrete encoders to map items into high-dimensional latent spaces. These raw embeddings lack distinct distributions, failing to accurately reflect item similarities or differences among different types of items (see left sub-figure of Figure 1), affecting precision of the generated results. Worse still, since sequence representation calculations depend on the item embeddings, it leads to high coupling, confusing the condition and potentially causing data to 'lose' its sampling direction in high-dimensional space. **(2) Slow Sampling.** Existing DM-based recommendation approaches suffer from slow inference, which hinders their practical usability. Slow sampling (i.e., generation) speed has long been an issue with DMs (Ho, Jain, and Abbeel 2020; Dhariwal and Nichol 2021). Studying this problem in the context of recommendation systems is crucial because large-scale user or item-based recommendation systems (such as Amazon[1] and Walmart[2]) demand high inference efficiency. Especially in online recommendation systems, the real-time nature of recommendations significantly impacts user experience.

To address the aforementioned issues, we propose **D**iffusion models enhanced with **A**uxiliary information and **E**fficient sampling mechanism **For** sequential **Rec**ommendation (**DAE4Rec**). The basic architectural differences between representative prior works and our approach are shown in the right sub-figure of Figure 1. In DAE4Rec, we design two modules: the Auxiliary Information Mining-Exploration (AIME) and the Efficient Sampling Mechanisms (ESM). To overcome the first challenge, the AIME module reconstructs the predefined graph using a Graph Auto Encoder (GAE) to obtain item latent representations that capture implicit global dependencies. We also design a matrix norm-based method, correcting item embeddings through the norm distance between the similarity measurement and the GAE-reconstructed graph, gently injecting auxiliary information into the DM. To address the second challenge, the ESM module accelerates the reverse sampling process by a non-Markov chain, achieving 'skip-step' sampling. Furthermore, we creatively designed a compensator to offset the performance degradation caused by the acceleration. The main contributions of the paper are summarized as:

- We incorporated auxiliary information from the graph structure into the DM using a GAE, enhancing the item embeddings and the personalization of recommendation.
- We are the first to implement an accelerated sampling process in DM-based recommendation, significantly improving sampling efficiency but causing minimal performance degradation with a designed compensator.
- Comprehensive experiments on three real-world datasets demonstrate the efficacy of our approach.

2 Related Work

In this section, we first review other competitive methods in generative SRs. Then, we briefly introduce the related work on DMs pertinent to this paper.

2.1 Generative Sequential Recommenders

GANs and VAEs are classical generative models, and a series of SR methods based on them have been proposed.

Methods based on GANs. GANs consist of a generator and a discriminator, trained together in a game-theoretic framework to produce realistic data. MFGAN (Ren et al.

[1]According to Amazon Statistics (2024), Amazon had approximately 310 million active users in 2023.

[2]According to the 60 Walmart Statistics (2024) – Users, Employees & Revenue, Walmart had over 240 million weekly visitors globally.

2020) uses a transformer-based generator for user behavior sequences and factor-specific discriminators. SSRGAN (Lv et al. 2021) tackles SR in streaming with a GAN-based negative sampling strategy to generate informative negative samples. Wang et al. proposed a dual adversarial network to align generated samples with true user preferences and generate samples that deviate from the objective to broaden the model's experience.

Methods based on VAEs. VAEs combine deep learning with variational inference, using an encoder to map data to a latent space and a decoder to reconstruct it, enabling data generation. Mult-VAE (Liang et al. 2018) is a key VAE-based recommendation method. SVAE (Sachdeva et al. 2019) enhances it by incorporating RNNs to capture temporal dependencies. VSAN (Zhao et al. 2021) integrates VAE with self-attention to address uncertainty and dynamics in user preferences. To improve posterior representation, ACVAE (Xie et al. 2021) adds adversarial training and contrastive loss within the AVB framework. ContrastVAE (Wang et al. 2022b) introduces a dual-branch VAE model based on contrastive learning to mitigate posterior collapse.

2.2 Guidance Diffusion and Accelerated Sampler

DMs are powerful generative models inspired by physical thermodynamics (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020; Song et al. 2020; Karras et al. 2022), which define a forward process to gradually corrupt data and a reverse process to generate data.

Guidance Diffusion. A series of conditional DMs (Rombach et al. 2022; Saharia et al. 2022) were proposed. Dhariwal and Nichol introduced classifier guidance, using a classifier's gradient on noisy data to adjust the DM sampling process towards a desired class label. Later, Ho and Salimans proposed classifier-free guidance, which combines predictions with and without conditions to balance sample quality and diversity.

Accelerated Sampling Process. Addressing the slow sampling in DMs, DDIM (Song, Meng, and Ermon 2020) accelerates sampling by using a non-Markovian approach, compared to DDPM (Ho, Jain, and Abbeel 2020), which requires many Markov chain steps. DPM-Solver (Lu et al. 2022a) improves upon this by decoupling the ODE solution proposed by Song et al. into linear and nonlinear parts, with DDIM as a first-order special case. DPM-Solver++ (Lu et al. 2022b) further refines it by addressing guidance diffusion issues.

3 Methods

In this section, we provide a detailed description of proposed DAE4Rec. We formulate the problem at first, after that show the framework of DAE4Rec, then explain the graph structure and how to introduce auxiliary information from it, and finally, we describe the efficient sampling mechanism employed. The model framework is shown in Figure 2.

3.1 Problem Statement

Given a set of users U and a set of items V, for any user $u \in U$, we denote their interaction sequence as $s_u = [v_1, v_2, \ldots, v_r, \ldots, v_n]$ with length $l_u = n$, where $v_r \in V$ represents the item of user u in their r-th interaction. The item v_n is the subsequent interacted item in the sequence (i.e. the target item). The task of SRs is to predict v_n, based on the given historical interaction sequence $v_{1:n-1} = [v_1, v_2, \ldots, v_r, \ldots, v_{n-1}]$.

From this, SRs can be viewed as exploring the conditional probability distribution of the target item $q(v_n|h)$ for each user (Yuan et al. 2019), where h represents the condition from historical interaction information in the sequences. Generally, let Enc denote a discrete data encoder that maps the data into a high-dimensional latent space, any $v \in V$ is represented as a corresponding embedding vector $e = \text{Enc}(v)$ and all vectors $E = \text{Enc}(V)$. With the condition derived from the embeddings in the historical interaction sequence, the problem can be reformulated as estimating $q(e_n|h(e_{1:n-1}))$, where $e_{1:n-1} = \text{Enc}(v_{1:n-1})$.

3.2 Model Framework

In this part, we will introduce the framework of our proposed DAE4Rec. Specifically, it includes explanations of the model backbone and the diffusion loss function we utilized.

Backbone of DAE4Rec. Given the target item embedding e_n^0 (e^0 for brevity), which have: $e^0 \sim q(e^0)$, the DM aims to learn a modeling distribution $p_\theta(e^0)$, enabling sampling of data that approximates the true data distribution.

Like classical DMs (Ho, Jain, and Abbeel 2020), we anchor a Markov chain through a series of latent variables $e^1, e^2, \ldots, e^T$ in the same sample space as e^0 through:

$$q(e^t|e^{t-1}) = \mathcal{N}(e^t; \sqrt{1-\beta_t}e^{t-1}, \beta_t I), \tag{1}$$

known as the *forward process* or *diffusion process*, where t is the diffusion timestep, a positive integer ranging from 1 to T. As shown by the gray arrows in the right sub-figure of Figure 2, Eq. (1) establishes a smooth transition from the data distribution to a pure Gaussian distribution $\mathcal{N}(e^T; 0, I)$, by progressively add Gaussian noise to the data e^0 according to a variance schedule $\beta_1, \beta_2, \ldots, \beta_T$ held constant as hyperparameters. One advantage of defining the forward process this way is that it allows closed-form sampling of e^t at any timestep t. Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, then it has $q(e^t|e^0) = \mathcal{N}(e^t; \sqrt{\bar{\alpha}_t}e^0, (1-\bar{\alpha}_t)I)$, which means:

$$e^t = \sqrt{\bar{\alpha}_t}e^0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{2}$$

where $\epsilon \sim \mathcal{N}(0, I)$ is sampled unit-noise.

Based on the inversion of the forward process, which is referred to as the *reverse process* or *sampling process*, DM iteratively samples e^0 begin with $\mathcal{N}(e^T; 0, I)$, denoted as: $p_\theta(e^{t-1}|e^t) = \mathcal{N}(\mu_\theta(e^t, t), \widetilde{\beta}_t I)$ in which $\mu_\theta(e^t, t)$ is:

$$\mu_\theta(e^t, t) = \sqrt{\bar{\alpha}_{t-1}}f_\theta(e^t, t) + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{\sqrt{1-\bar{\alpha}_t}}\epsilon, \tag{3}$$

where $f_\theta(e^t, t)$ is the MLP to predict e^0 by e^t, and the second term is used to match the variance in the forward process.

As mentioned, the final aim is to estimate the conditional probability distribution of the target item embedding vectors by $p_\theta(e_n^0|h(e_{1:n-1}))$ i.e, $p_\theta(e^0|h)$, under specified condition

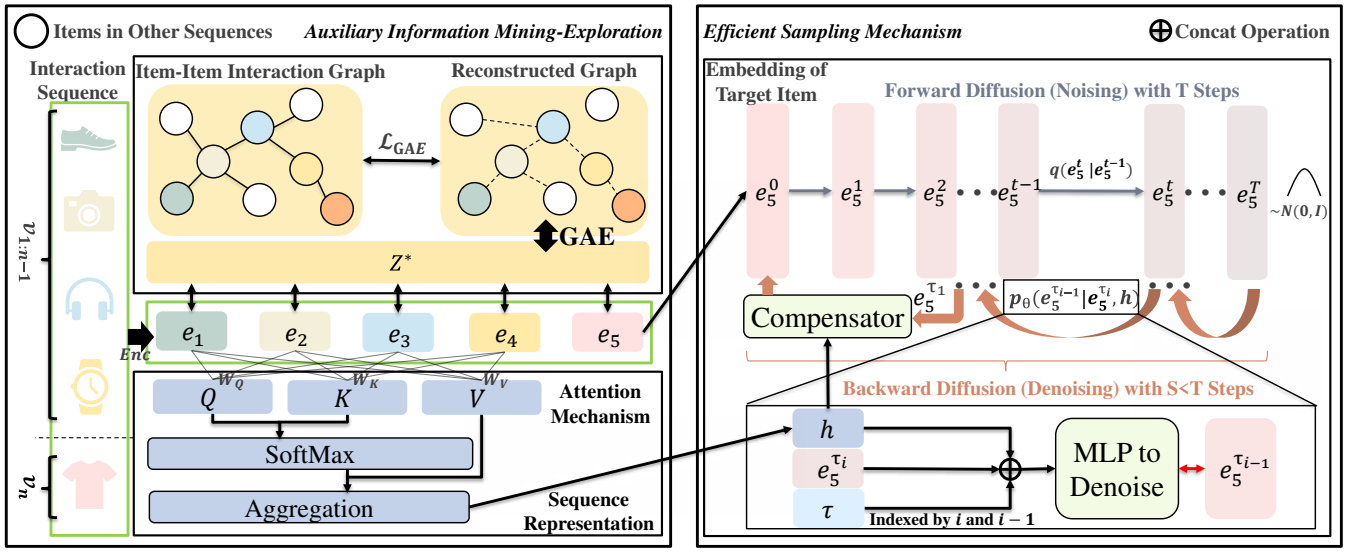


Figure 2: The proposed DAE4Rec framework includes an auxiliary information mining-exploration module (left sub-figure) and an efficient sampling mechanism module (right sub-figure). For a user's interaction sequence (leftmost green box), item embeddings are obtained using an encoder *Enc* and combined with node latent representations learned by the GAE. The target item is treated as the generated object, with the historical interaction sequence serving as input to the attention mechanism, producing sequence representation h as the diffusion guidance signal. During generation, accelerated target item sampling is achieved through a non-Markov chain (curved arrows in the right sub-figure), with residuals mitigated by the compensator.

h. Compared to previous work, we use a simplified attention mechanism to calculate the condition:

$$\text{Attention}(e_{1:n-1}) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{4}$$

where $[Q, K, V] = e_{1:n-1}[W_Q, W_K, W_V]$ and W_Q, W_K, W_V represent weight matrices, d_k is the dimension of the key vector used to scale the dot product result. Subsequently, we sum the attention outputs along the sequence dimension to obtain the aggregated condition h.

Diffusion loss function. To achieve personalized target item distribution estimation through h. The sampling process $p_\theta(e^{t-1}|e^t, h)$ ought to be guided by sequential information as the condition. We achieve this by concatenating the inputs to the MLP f_θ, so the diffusion training objective becomes:

$$\mathcal{L}_{\text{Diff}} = \mathbb{E}_{e^0,\epsilon}\left[\lambda_t \|e^0 - f_\theta(\sqrt{\bar{\alpha}_t}e^0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t, h)\|^2\right], \tag{5}$$

where $\lambda_t = \bar{\alpha}_{t-1}(1-\bar{\alpha}_t)/2\beta_t(1-\bar{\alpha}_{t-1})$ and constant terms are omitted.

3.3 Auxiliary Information Mining-Exploration

Guidance diffusion leverages meaningful sequential information for target item generation, while previous work focuses solely on user preferences, neglecting underlying item features. This section introduces how to explore auxiliary information and enhance item embeddings in the DM. Since item categorical features are implicit in their global adjacency relationships (Wu et al. 2019) (e.g., if item B is adjacent to item A in a sequence, A is likely more similar to B than a non-adjacent item C), we focus on adjacency dependencies across interaction sequences. Our goal is to ensure item embedding similarities accurately reflect this information (as shown in the left sub-figure of Figure 1) and integrate it into DM training. We achieve this in three steps: (1) constructing a graph to model auxiliary information; (2) mining this information via a GAE; and (3) incorporating it into diffusion training through a tailored optimization objective.

Item-Item Interaction Graph Establishment. Firstly, to capture the dependencies among user interaction sequences, we construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over all items. Here, $\mathcal{V}$ is the vertex set derived from the set of items V. For the edge set $\mathcal{E}$, we traverse all s_u and establish an edge between two adjacent items in any sequence considering duplicate edges only once. Formally described as:

$$\mathcal{E} = \{(v_i, v_{i+1}) \text{ in } s_u, u \in U, 1 \le i \le l_u - 1\}. \tag{6}$$

Such a graph structure models dependencies between items from a global perspective, rather than a single sequence. Given that it essentially considers relationships among items, we name it as *Item-Item Interaction Graph* $\mathcal{G}$.

Auxiliary Information Mining. Then, we aim to obtain interpretable latent representations of item nodes in $\mathcal{G}$. A GAE is well-suited for this purpose, as it can incorporate node features using unsupervised learning methods.

To illustrate the mentioned GAE, we introduce the adjacency matrix A of $\mathcal{G}$, where the diagonal elements are set to 1, and its degree matrix D, where the diagonal element $D_{ii} = \sum_i A_{ij}$ and others are 0. We obtain the latent representation $Z = [z_1, z_2, \ldots, z_v, \ldots, z_{|V|}]$ of item nodes

through the GAE consists of a two-layer Graph Convolutional Network (GCN), defined as follows:

$$Z = \widetilde{A}\,\mathrm{ReLU}\big(\widetilde{A}W_0\big)W_1, \tag{7}$$

where $\widetilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is normalized adjacency matrix, $\mathrm{ReLU}(\cdot)$ is activation function, and W_0 and W_1 are learnable weight matrices for the two-layer GCN respectively.

Next, we choose the cosine similarity, which is simple but effective. The similarity matrix calculated as follows:

$$S = \sigma(Z^T Z). \tag{8}$$

That is, for item i and item j, the similarity in S is the element $S_{ij} = \sigma(z_i^T z_j)$, and $\sigma(\cdot)$ is the logistic sigmoid function. For measuring the similarity in the latent representation to reconstruct the dependencies between items inherent in $\mathcal{G}$, we assume: $p\,(\hat{A}_{ij} = 1|z_i, z_j) = S_{ij}$, where $\hat{A}$ is the adjacency matrix of reconstructed graph, and more directly, there is: $\hat{A}_{ij} = S_{ij}$. Practically, the Binary Cross Entropy (BCE) is used to train GAE:

$$\mathcal{L}_{\text{GAE}} = -\frac{1}{|V|}\sum_{i=1}^{|V|}\big[A_i^T \log(\hat{A}_i) + (1 - A_i^T)\log(1 - \hat{A}_i)\big], \tag{9}$$

where A_i and $\hat{A}_i$ are the ground-truth adjacency vector and the reconstructed adjacency vector of item node i respectively. And $\log(\cdot)$ denotes the logarithm on a vector.

The process of pretraining the GAE using Eq. (9) is necessary, the similarity measure can be employed to model the prior auxiliary dependencies of items in the graph $\mathcal{G}$ within the latent embedding. After that, we fix it into the auxiliary information exploration. For ease of explanation, we denote the well-trained GAE latent representations as Z^*.

Auxiliary Information Exploration. Intuitively, it is tempting to use Z^* as the initial embedding vectors E to train the DM. However, due to the strict training objective in Eq. (9), this approach would disrupt the already learned knowledge.

Instead, we plan to devise an alternative strategy that dynamically learns from Z^*, incorporates it with the embedding vectors learned during the diffusion process. We design the following loss function based on the matrix norm:

$$\mathcal{L}_{\text{AIME}} = \|\sigma(E^T E) - \sigma(Z^{*T} Z^*)\|^2, \tag{10}$$

where $\sigma(E^T E)$ denotes the similarity among different items from diffusion embeddings, and $\sigma(Z^{*T} Z^*)$ from GAE.

Intuitively, Eq. (10) achieves aligning representation at the level of reconstructing the graph. It has its rationale and benefits: (1) Firstly, when Z^* and E have the same latent dimension, the reconstructed adjacency matrix $\hat{A}$ is easier to fit compared to A; (2) Additionally, the norm is more relaxed compared to element-wise operations, allowing the modeling of item dependencies in the graph while ensuring that the embeddings learned by the DM remain largely intact. Combining it with the loss term we adopted from Eq. (5), we present the loss function of DAE4Rec:

$$\mathcal{L} = \mathcal{L}_{\text{Diff}} + \delta \cdot \mathcal{L}_{\text{AIME}}, \tag{11}$$

where δ controls the importance of AIME loss term $\mathcal{L}_{\text{AIME}}$.

3.4 Efficient Sampling Mechanism

In this section, we detail the efficient sampling mechanism proposed. Specifically, it contains the non-Markovian anchored sampling process and an error fitting network.

The non-Markovian anchored Sampling Process. Recall the reverse process $p_\theta(e^{t-1}|e^t)$ requires T steps to generate samples. We employ a class of non-Markov chains to reduce the number of iterations in the reverse process.

To achieve this goal, we rely on a series of distributions $q(e^t|e^{t-1}, e^0)$ as the new forward process. At this point, the forward process is no longer a Markov chain, as each e^t depends on both e^0 and e^{t-1}. Without loss of generality, we assume a joint distribution anchored by such a process is:

$$q(e^{t-1}|e^t, e^0) = \mathcal{N}(\kappa_t e^t + \xi_t e^0, \rho_t^2 I), \tag{12}$$

where κ_t, ξ_t and ρ_t are coefficients to be determined. To maintain the marginal distributions at each time step, thereby without altering the model training achieved through Eq. (11), we get e^t and e^{t-1} by Eq. (2). Furthermore, by e^{t-1} sampled from the joint distribution defined in Eq. (12), we can derive the coefficients to be determined:

$$\begin{aligned}\kappa_t &= \sqrt{\frac{1 - \bar{\alpha}_{t-1} - \rho_t^2}{1 - \bar{\alpha}_t}},\\ \xi_t &= \sqrt{\bar{\alpha}_{t-1}} - \sqrt{\frac{\bar{\alpha}_t(1 - \bar{\alpha}_{t-1} - \rho_t^2)}{1 - \bar{\alpha}_t}},\end{aligned} \tag{13}$$

where ρ_t is variable parameters. In this paper, we set $\rho_t = 0$ for all t like Song, Meng, and Ermon do.

Based on Eq. (13), the sampling results from the joint distribution defined in Eq. (12) can be organized as follows:

$$e^{t-1} = \sqrt{\bar{\alpha}_{t-1}}e^0 + \sqrt{1 - \bar{\alpha}_{t-1}}\big(\frac{e^t - \sqrt{\bar{\alpha}_t}e^0}{\sqrt{1 - \bar{\alpha}_t}}\big). \tag{14}$$

In the previous discussion, the generation process is an approximate inversion of the forward process, which has T steps, thus requiring also T steps of sampling. Now we consider a smaller sampling step $S < T$ to accelerate the generation process by the sampling defined in Eq. (14).

Let τ be a subsequence of length S from $[1, \ldots, T]$, where $S < T$ and $i \in [S]$ is the index of τ. For the chain $\{e^{\tau_i}\}_{i=1}^S$ starting from $p_\theta(e^{\tau_S}) = \mathcal{N}(0, I)$, we set $p_\theta(e^{\tau_{i-1}}|e^{\tau_i}, h) = q(e^{\tau_{i-1}}|e^{\tau_{i-1}}, e^0, h)$, which means:

$$e^{\tau_{i-1}} = \sqrt{\bar{\alpha}_{\tau_{i-1}}}f_\theta^{(\tau_i)} + \sqrt{1 - \bar{\alpha}_{\tau_{i-1}}}\frac{e^{\tau_i} - \sqrt{\bar{\alpha}_{\tau_i}}f_\theta^{(\tau_i)}}{\sqrt{1 - \bar{\alpha}_{\tau_i}}}, \tag{15}$$

where $i > 1$ and $f_\theta^{(\tau_i)} = f_\theta(e^{\tau_i}, \tau_i, h)$ is the denoise MLP to predict e^0 from e^{τ_i}. Note that we directly incorporated guidance condition h into the sampling process and the final step for $e^{\tau_1} \to e^0$ will be discussed in following part.

As illustrated by the curved arrows on the right sub-figure of Figure 2, this sampling process is akin to a 'skip-step' across T timesteps.

Compensator for the Sampling Process. The sampling process introduces a gap between the true and modeled distributions, causing performance degradation. Specifically,

Methods	Foursquare				Gowalla				Retailrocket			
	HR@		NDCG@		HR@		NDCG@		HR@		NDCG@	
	10(%)	20(%)	10(%)	20(%)	10(%)	20(%)	10(%)	20(%)	10(%)	20(%)	10(%)	20(%)
Caser	17.69	24.77	9.78	11.56	9.32	13.15	5.27	6.23	12.48	16.49	7.45	8.26
SASRec	<u>38.93</u>	<u>48.65</u>	<u>25.03</u>	<u>27.50</u>	12.71	15.08	8.27	8.86	31.14	34.54	22.08	22.95
STOSA	19.23	24.92	9.98	11.41	9.12	13.06	5.21	6.20	11.83	16.16	6.74	7.84
ACVAE	16.56	22.75	9.83	11.43	10.08	11.25	5.21	5.48	10.96	13.72	5.83	6.24
ContrastVAE	14.21	19.99	6.84	8.28	6.24	9.16	3.52	4.25	7.54	10.98	3.99	4.86
DiffRec	18.62	24.80	10.70	11.80	6.72	9.61	5.54	6.61	10.28	11.65	4.90	5.28
DreamRec	20.25	27.25	19.72	25.18	<u>21.59</u>	<u>22.57</u>	<u>19.50</u>	<u>19.77</u>	<u>38.19</u>	<u>39.64</u>	<u>28.64</u>	<u>29.01</u>
DAE4Rec	**39.88**	**48.98**	**26.02**	**28.32**	**23.99**	**25.50**	**21.29**	**21.67**	**39.46**	**45.75**	**30.36**	**31.95**

Table 1: The performance of DAE4Rec with other baselines over three datasets. Bold font indicates the best performance, while underlining indicates the second-best. In the table, DAE4Rec uses a sampling step size S of half the diffusion timestep T. The reported metrics are the averages obtained from three independent runs.

treating the MLP output in Eq. (15) as an unbiased estimate introduces bias (Hang et al. 2023), which is further amplified by 'skip-step' sampling, leading to deviations in the sampling trajectory (Kim and Ye 2022).

However, we designed a unique network c_ϕ, referred to as the *compensator* to relieve this issue in the final sampling step. We consider this after training f_θ. Based on the well-trained MLP f_θ^*, we sample e^{τ_1} by Eq. (15). Subsequently, we train the compensator c_ϕ using the following objective:

$$\mathcal{L}_{\text{Com}} = \|e^0 - c_\phi(e^{\tau_1}, h)\|^2. \tag{16}$$

Then, the final step is $\hat{e}^0 = c_\phi(e^{\tau_1}, h)$ instead of $\hat{e}^0 = f_\theta^{*(\tau_1)}$.

Finally, the score for each item is calculated through inner products with generated $\hat{e}^0$. The top-k items are determined based on their ranking, forming the recommendation list.

4 Experiments and Analysis

In this section, we first briefly introduce the datasets used in the experiments. Then, we compare DAE4Rec with other methods and we validate the effectiveness of its components. Finally, we discuss and analyze the critical hyperparameters.

4.1 Datasets

We selected three real-world recommendation datasets, their statistics are summarized in Table 2:

- **Foursquare** (Levandoski et al. 2012; Sarwat et al. 2014): A location-based service dataset containing user check-in data, with venue IDs used as recommendation targets.
- **Gowalla** (Cho, Myers, and Leskovec 2011): A geolocation-based dataset from the social networking site Gowalla, utilizing user check-in data.
- **Retailrocket** (Zykov, Artem, and Alexander 2022): An implicit feedback dataset from an e-commerce platform on Kaggle, utilizing user behavior data.

Statistics	Foursquare	Gowalla	Retailrocket
Records	303.0K	934.1K	549.0K
Users	40.5K	51.6K	59.5K
Items	17.2K	33.3K	27.5K
Average sequence length	7.47	18.10	9.23
Sparsity	99.96%	99.95%	99.97%

Table 2: Statistics of datasets used in the experiments.

4.2 Main Results

The comparison of performance between DAE4Rec and baselines under top-10 and top-20 settings is shown in Table 1. Experimental results show that DAE4Rec achieves the best performance across all evaluated metrics on the three datasets. This validates its superiority.

For sequence learning models like Caser, SASRec, and STOSA, result shows that their performance is relatively average. High data sparsity hinders the convolutional layers and GRUs from effectively capturing sequential features.

For generative models, the methods based on DM significantly outperform those based on VAE. Our proposed DAE4Rec achieved the best performance, largely due to the auxiliary item information integration approach we introduced and the compensator's ability to correct residuals in efficient sampling. We further validate the effectiveness of these components in the following subsection.

4.3 Ablation Study

In this part, we validate the effectiveness of the components for improving recommendation performance in DAE4Rec, achieved through comparisons with DreamRec and three variants of DAE4Rec: -C, -C_{ground} and -A . The experimental results on Foursquare are shown in Table 3.

Effect of the Auxiliary Information. The auxiliary information mining-exploration module is a key component of DAE4Rec. To assess its impact, we remove it from variant -C (i.e., $\delta = 0$ in DAE4Rec). In variant -C_{ground}, we replace

AIME	Compensator	Model	HR@		NDCG@	
			10(%)	20(%)	10(%)	20(%)
✗	✗	DreamRec	20.25	27.25	19.72	25.18
✗	✓	-C	27.53	29.50	23.57	24.67
✗	✓	-C_{ground}	31.71	33.36	24.95	25.38
✓	✗	-A	37.10	45.14	25.46	26.97
✓	✓	DAE4Rec	**39.88**	**48.98**	**26.02**	**28.32**

Table 3: Model ablation study with variants and DreamRec on Foursquare. Here, the options AIME and Compensator respectively indicate whether the auxiliary information mining-exploration module and sampling compensator were used.

the reconstructed graph target with the ground-truth graph derived from sequence data to verify the benefit of introducing prior information through Eq. (10). Results show a significant performance drop in variant -C. Although variant -C_{ground} performs better than -C, it still has limitations, likely because the non-binary reconstructed graph target in DAE4Rec contains richer and smoother item dependency information. Additionally, under the relaxed constraints of a matrix norm-based optimization objective, it is easier to learn for embeddings with limited dimensions.

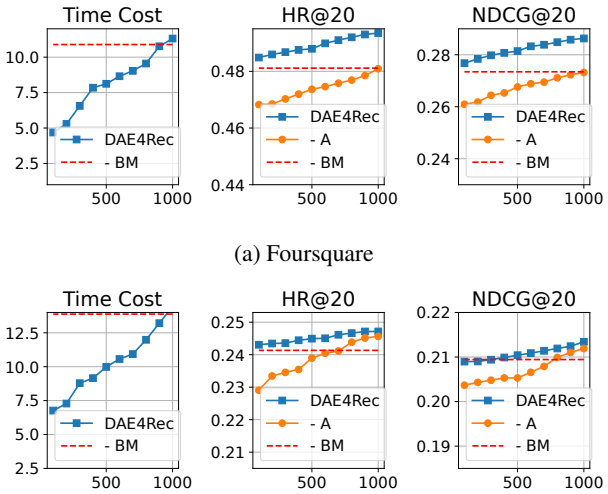
Effect of the Proposed Sampling Compensator. The third variant, -A, uses the same auxiliary information as DAE4Rec but omits the compensator from the sampling mechanism. Results show that -A underperforms, highlighting the necessity of the compensator, which optimizes the sampling trajectory to enhance recommendation quality.

4.4 Hyperparameter Study

In this part, we conduct experimental studies on the impact on several key hyperparameters in DAE4Rec. The results are shown in Figure 3 and Figure 4.

Sampling Step Size S. It controls the number of steps DAE4Rec iteratively samples the target item using the efficient sampling mechanism. In Figure 3, we use the model with 1000 diffusion timesteps (i.e., T) and the Markov chain-based sampling method as a benchmark (i.e., -BM in the figures) to explore the performance and time overhead under different S settings in DAE4Rec and variant -A. The results indicate: (1) The efficient sampling mechanism aligns well with DAE4Rec, significantly reducing inference time while maintaining excellent performance; (2) As S decreases, performance degradation is intuitive, but the compensator effectively mitigates residuals, resulting in less than 4% overall performance degradation in our experiments; (3) Comparing results across datasets reveals that on those with longer interaction sequences, 'skip-step' introduces larger residuals, making the compensator particularly important.

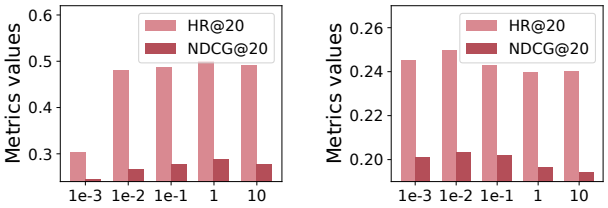
Weight of AIME Loss Term δ. This hyperparameter controls the importance of the AIME loss term $\mathcal{L}_{\text{AIME}}$ during training. In Figure 4, we explore performance of DAE4Rec under different δ settings, using powers of 10 to emphasize differences. The results demonstrate the necessity of the $\mathcal{L}_{\text{AIME}}$, as performance deteriorates significantly with a too small δ. However, with a further increase in δ, performance declines, likely due to excessive focus on auxiliary information, which weakens the learning of attention mechanism and denoise MLP f_θ, affecting both personalization and generation quality. Additionally, across datasets, smaller δ values be required for those with longer sequences.



(a) Foursquare

(b) Gowalla

Figure 3: The model performance under different S. Diffusion step T in all models are set to 1000.



(a) Foursquare (b) Gowalla

Figure 4: The model performance under different δ.

5 Conclusions

In this paper, we address the shortcomings of previous diffusion model-based sequential recommenders, specifically the highly coupled representations and slow sampling speed. To overcome these issues, we propose a method that incorporates auxiliary information via a GAE and a more efficient sampling mechanism, enabling the modeling of more distinguishable and less error-prone conditions with faster inference speeds. Additionally, we designed a compensator, effectively mitigating errors from accelerated sampling. Empirical results demonstrate the superior performance of our proposed method in both accuracy and efficiency.

Acknowledgments

This work was supported by the Natural Science Foundation of Shandong Province (No. ZR2023MF007), National Natural Science Foundation of China (No. 92267104, No. 62372242), State Key Laboratory of New Software Technology Open Project (No. KFKT2024B50), Foundation of Yunnan Key Laboratory of Service Computing (No. YNSC23103). Dr. Xuyun Zhang is the recipient of an ARC DECRA (project No. DE210101458) funded by the Australian Government.

References

Chen, X.; Xu, H.; Zhang, Y.; Tang, J.; Cao, Y.; Qin, Z.; and Zha, H. 2018. Sequential recommendation with user memory networks. In *WSDM '18*, 108–116.

Cho, E.; Myers, S. A.; and Leskovec, J. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1082–1090.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Hang, T.; Gu, S.; Li, C.; Bao, J.; Chen, D.; Hu, H.; Geng, X.; and Guo, B. 2023. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7441–7451.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, 173–182.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.

Karras, T.; Aittala, M.; Aila, T.; and Laine, S. 2022. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35: 26565–26577.

Kim, B.; and Ye, J. C. 2022. Denoising mcmc for accelerating diffusion-based generative models. *arXiv preprint arXiv:2209.14593*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Levandoski, J. J.; Sarwat, M.; Eldawy, A.; and Mokbel, M. F. 2012. Lars: A location-aware recommender system. In *2012 IEEE 28th international conference on data engineering*, 450–461. IEEE.

Li, X.; Thickstun, J.; Gulrajani, I.; Liang, P. S.; and Hashimoto, T. B. 2022. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35: 4328–4343.

Li, Z.; Sun, A.; and Li, C. 2023. Diffurec: A diffusion model for sequential recommendation. *ACM Transactions on Information Systems*, 42(3): 1–28.

Liang, D.; Krishnan, R. G.; Hoffman, M. D.; and Jebara, T. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, 689–698.

Liu, W.; Chen, C.; Liao, X.; Hu, M.; Su, J.; Tan, Y.; and Wang, F. 2024a. User Distribution Mapping Modelling with Collaborative Filtering for Cross Domain Recommendation. In *Proceedings of the ACM on Web Conference 2024*, 334–343.

Liu, W.; Chen, C.; Liao, X.; Hu, M.; Tan, Y.; Wang, F.; Zheng, X.; and Ong, Y. S. 2024b. Learning Accurate and Bidirectional Transformation via Dynamic Embedding Transportation for Cross-Domain Recommendation. In *AAAI '24*, volume 38, 8815–8823.

Liu, W.; Su, J.; Chen, C.; and Zheng, X. 2021. Leveraging distribution alignment via stein path for cross-domain cold-start recommendation. *Advances in Neural Information Processing Systems*, 34: 19223–19234.

Liu, W.; Zheng, X.; Chen, C.; Su, J.; Liao, X.; Hu, M.; and Tan, Y. 2023a. Joint internal multi-interest exploration and external domain alignment for cross domain sequential recommendation. In *Proceedings of the ACM Web Conference 2023*, 383–394.

Liu, W.; Zheng, X.; Hu, M.; and Chen, C. 2022a. Collaborative filtering with attribution alignment for review-based non-overlapped cross domain recommendation. In *Proceedings of the ACM web conference 2022*, 1181–1190.

Liu, W.; Zheng, X.; Su, J.; Hu, M.; Tan, Y.; and Chen, C. 2022b. Exploiting variational domain-invariant user embedding for partially overlapped cross domain recommendation. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, 312–321.

Liu, W.; Zheng, X.; Su, J.; Zheng, L.; Chen, C.; and Hu, M. 2023b. Contrastive proxy kernel stein path alignment for cross-domain cold-start recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(11): 11216–11230.

Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022a. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35: 5775–5787.

Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022b. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*.

Lv, Y.; Xu, J.; Zhou, R.; Fang, J.; and Liu, C. 2021. SSRGAN: A generative adversarial network for streaming sequential recommendation. In *DASFAA 2021*, 36–52. Springer.

Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Ren, R.; Liu, Z.; Li, Y.; Zhao, W. X.; Wang, H.; Ding, B.; and Wen, J.-R. 2020. Sequential recommendation with self-attentive multi-adversarial network. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 89–98.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Sachdeva, N.; Manco, G.; Ritacco, E.; and Pudi, V. 2019. Sequential variational autoencoders for collaborative filtering. In *Proceedings of the twelfth ACM international conference on web search and data mining*, 600–608.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494.

Sarwat, M.; Levandoski, J. J.; Eldawy, A.; and Mokbel, M. F. 2014. Lars*: An efficient and scalable location-aware recommender system. *IEEE Transactions on Knowledge & Data Engineering*, 26(06): 1384–1399.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2256–2265. PMLR.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

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 the 28th ACM international conference on information and knowledge management*, 1441–1450.

Tevet, G.; Raab, S.; Gordon, B.; Shafir, Y.; Cohen-Or, D.; and Bermano, A. H. 2022. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*.

Wang, F.; Chen, C.; Liu, W.; Fan, T.; Liao, X.; Tan, Y.; Qi, L.; and Zheng, X. 2024a. CE-RCFR: Robust counterfactual regression for consensus-enabled treatment effect estimation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3013–3023.

Wang, F.; Liu, W.; Chen, C.; Zhu, M.; and Zheng, X. 2022a. HCFRec: Hash Collaborative Filtering via Normalized Flow with Structural Consensus for Efficient Recommendation. *arXiv preprint arXiv:2205.12042*.

Wang, S.; Hu, L.; Wang, Y.; Cao, L.; Sheng, Q. Z.; and Orgun, M. 2019. Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830*.

Wang, W.; Xu, Y.; Feng, F.; Lin, X.; He, X.; and Chua, T.-S. 2023. Diffusion recommender model. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 832–841.

Wang, Y.; Liu, Z.; Yang, L.; and Yu, P. S. 2024b. Conditional denoising diffusion for sequential recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 156–169. Springer.

Wang, Y.; Zhang, H.; Liu, Z.; Yang, L.; and Yu, P. S. 2022b. Contrastvae: Contrastive variational autoencoder for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2056–2066.

Wang, Z.; Ye, W.; Chen, X.; Zhang, W.; Wang, Z.; Zou, L.; and Liu, W. 2022c. Generative session-based recommendation. In *Proceedings of the ACM Web Conference 2022*, 2227–2235.

Wu, C.-Y.; Ahmed, A.; Beutel, A.; Smola, A. J.; and Jing, H. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, 495–503.

Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; and Tan, T. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 346–353.

Xie, Z.; Liu, C.; Zhang, Y.; Lu, H.; Wang, D.; and Ding, Y. 2021. Adversarial and contrastive variational autoencoder for sequential recommendation. In *Proceedings of the Web Conference 2021*, 449–459.

Yang, Z.; He, X.; Zhang, J.; Wu, J.; Xin, X.; Chen, J.; and Wang, X. 2023. A generic learning framework for sequential recommendation with distribution shifts. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 331–340.

Yang, Z.; Wu, J.; Wang, Z.; Wang, X.; Yuan, Y.; and He, X. 2024. Generate What You Prefer: Reshaping Sequential Recommendation via Guided Diffusion. *Advances in Neural Information Processing Systems*, 36.

Yuan, F.; Karatzoglou, A.; Arapakis, I.; Jose, J. M.; and He, X. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*, 582–590.

Zhao, J.; Zhao, P.; Zhao, L.; Liu, Y.; Sheng, V. S.; and Zhou, X. 2021. Variational self-attention network for sequential recommendation. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 1559–1570. IEEE.

Zhao, S.; Song, J.; and Ermon, S. 2019. Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, 5885–5892.

Zykov, R.; Artem, N.; and Alexander, A. 2022. Retailrocket recommender system dataset.