Learning Multiple User Distributions for Recommendation via Guided Conditional Diffusion

Cheng Wu[1*], Liang Su[1*], Chaokun Wang[1†], Shaoyun Shi[2], Ziqian Zhang[1], Ziyang Liu[1], Wang Peng[2], Wenjin Wu[2], Peng Jiang[2]

[1]School of Software, BNRist, Tsinghua University, Beijing 100084, China,
[2]Kuaishou Inc., Beijing 100000, China,
{wuc22, sul21, zhangzq22, liu-zy21}@mails.tsinghua.edu.cn, chaokun@tsinghua.edu.cn,
{shishaoyun, wangpeng16, wuwenjin, jiangpeng}@kuaishou.com

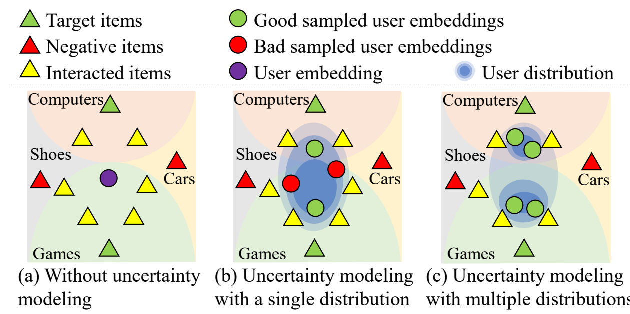
Abstract

Recommender systems are increasingly prevalent to provide personalized suggestions and enhance user satisfaction. Typical recommendation models encode users and items as embeddings, and generate recommendations by assessing the similarity between these embeddings. Despite their effectiveness, these embedding-based models struggle with modeling user uncertainty and capturing diverse user interests using a single fixed user embedding. Recent studies have begun to explore a user-distribution paradigm to learn distributions for users. However, this approach employs a single distribution per user, which fails to effectively delineate semantic boundaries, resulting in sub-optimal recommendations. To this end, we propose GCDR, a **G**uided **C**onditional **D**iffusion **R**ecommender model, to learn multiple distributions for each user in this paper. Specifically, GCDR addresses two major challenges: 1) learning disentangled distributions, and 2) learning personalized distributions. GCDR captures inter-user and intra-user distribution properties through conditional and guided diffusion, respectively. It maintains user-specific embeddings to encode long-term interests for conditional diffusion, while for guided diffusion, it incorporates short-term interests encoded from recent interactions with category preferences. To align the diffusion model with the recommendation task, we train GCDR with three losses, included the user loss, the recommendation loss and the diffusion loss. Extensive experiments on four real-world datasets show that GCDR is able to learn effective user distributions and is superior to thirteen state-of-the-art baseline methods.

Code — https://github.com/NoMultiply/GCDR

1 Introduction

Nowadays, recommender systems have become ubiquitous across a wide range of applications (Altenburger and Ho 2019; Gu et al. 2020; Chang et al. 2021; Liu et al. 2022). They incorporate recommendation models to provide personalized suggestions, thus enhancing user experience and satisfaction. Various recommendation models have been proposed, evolving from traditional Matrix Factorization (MF) (He et al. 2016) to innovative Deep Neural Network (DNN) (Sedhain et al. 2015; Wang et al. 2019). These models represent users and items as embeddings (*i.e.*, vectors), and make recommendations based on the similarity (e.g., cosine similarity) between these embeddings.



(a) Without uncertainty modeling (b) Uncertainty modeling with a single distribution (c) Uncertainty modeling with multiple distributions

Figure 1: Illustration of different user modeling approaches. (a) Without uncertainty modeling, the learned user embedding tends to be closest to the interacted items. This proximity makes it challenging to distinguish between target items and negative items. (b) When modeling the user as a distribution, we can sample good embeddings that are close to the target items. However, it is possible to sample bad embeddings that are near to negative items. (c) By modeling a user as multiple distributions, such as learning a separate distribution for each category of interacted items, it becomes easy to obtain embeddings close to the target items and far from the negative items.

While embedding-based recommendation models have shown impressive performance on recommendation effectiveness and efficiency (He et al. 2020; Gu et al. 2022; Wu et al. 2023a,b; Xu et al. 2023), the **user-embedding** paradigm, *i.e.*, learning a single and fixed user embedding, encounters difficulties in user uncertainty modeling, and fails to represent the diverse interest of users well (Li, Sun, and Li 2023; Wu et al. 2024). For example, in Figure 1(a), it is hard to discern whether the learned user embedding is closer to the target items from categories "Computers" and "Games", or to the negative items from categories "Shoes" and "Cars". To address this issue, several latest research efforts (Wu et al. 2024; Yang et al. 2024) shift towards a **user-**

*These authors contributed equally.
†Corresponding author.

distribution paradigm. These approaches involve learning a distribution for a user, and modeling user uncertainty by sampling different embeddings from this distribution. As illustrated in Figure 1(b), the good user embedding samples (*i.e.*, green circles) enable us to recommend target items rather than negative items.

Nevertheless, existing methods with user-distribution paradigm mostly characterize user uncertainty with a single distribution, which obscures the semantic boundaries (such as category boundaries) of user embeddings, and can result in bad user embedding samples (*i.e.*, red circles in Figure 1(b)). In this regard, we propose to learn **multiple distributions** for each user. As shown in Figure 1(c), by learning two distributions within the embedding sub-spaces of both the "Computer" category and the "Game" category, intentional and guided embedding sampling can be achieved, which helps to generate high quality user embeddings that are close to the target items.

However, modeling user uncertainty with multiple distributions is non-trivial, and there are two major challenges:

How to learn disentangled distributions. Disentangling the distributions of a user into different embedding subspaces is challenging. On the one hand, simply learning multiple distribution with multiple existing generative recommender models (such as GAN-based (Guo et al. 2020) and VAE-based (Ma et al. 2019) models) cannot guarantee the distinguishability of the learned distributions due to the lack of guidance. On the other hand, learning a distribution for every category weakens the connections of the learned distributions, and thus fails to capture the long-term consistency of a user. Therefore, it is necessary to explore a new approach that simultaneously considers both the individual distinctiveness and overall interdependence of distributions.

How to learn personalized distributions. Capturing user personality, such as long-term and short-term interest as well as category preference, is critical for recommendation models. Long-term interests often reflect stable, enduring preferences that can be integral to understanding overall user behavior, while short-term interests capture more transient, immediate needs and desires. Additionally, users may have distinct preferences across different categories. The challenge lies in effectively integrating these diverse dimensions of user behavior into multiple distributions that are capable of balancing the stability of long-term patterns with the dynamism of short-term fluctuations, all while maintaining sensitivity to categorical preferences.

To tackle the above challenges, we propose GCDR, a **G**uided **C**onditional **D**iffusion **R**ecommender model, to learn user uncertainty with multiple distributions. Specifically, GCDR captures the intra-user and inter-user distribution properties via guided diffusion and conditional diffusion, respectively. To learn disentangled distributions, GCDR performs diffusion conditioning on users to ensure the overall consistency of distributions for each user, and meanwhile guides the diffusion with different user characteristics to generate different distributions for each user. To learn personalized distributions, a user-specific embedding is maintained to encode long-term interest and serve as the condition for GCDR to distinguish different users, while the historical interaction sequence and the category preference are depicted as the guidance to enable intentional embedding sampling. Furthermore, we design three losses, namely the user loss, the recommendation loss, and the diffusion loss, to align the diffusion model with the recommendation task, and to train GCDR effectively.

The main contributions of this paper are as follows:

- We point out the limitations of the user-distribution paradigm, and first propose to model user uncertainty with multiple distributions for each user. (Sections 2 and 3).
- We develop GCDR, a novel diffusion recommender model that leverages guided conditional diffusion to learn disentangled and personalized user distributions. (Section 4).
- We conduct comprehensive experiments to evaluate the GCDR model with thirteen state-of-the-art baseline methods. The experimental results show that GCDR is superior to other methods, and is able to perform effective recommendation with the help of guided conditional diffusion. (Section 5).

2 Related Work

In this section, we review the related work relevant to this study, including *Generative Recommendation* and *Diffusion Recommendation*.

2.1 Generative Recommendation

Generative recommendation (Li, Kawale, and Fu 2015; Li and She 2017; Yu et al. 2019; Ren et al. 2020; Shenbin et al. 2020) has become a pivotal approach in recommendation systems, offering enhanced capabilities to model intricate user-item interactions. The landscape of classical generative recommendation is divided mainly into GAN-based (Guo et al. 2020; Jin et al. 2020; Gao et al. 2021; Xu et al. 2022; Chen et al. 2022) and VAE-based methods (Zhang, Yao, and Xu 2017; Ma et al. 2019; Liu et al. 2021). GAN-based techniques employ adversarial training frameworks where a generator predicts user interactions while a discriminator evaluates these predictions, iteratively refining the accuracy (Wang et al. 2017; He et al. 2018; Wu et al. 2019; Wang et al. 2022b). VAE-based methods, on the other hand, focus on encoding user data into a latent space and then decoding it to reconstruct interaction probabilities (Higgins et al. 2017; Liang et al. 2018; Shenbin et al. 2020; Truong, Salah, and Lauw 2021; Xie et al. 2021; Salah, Tran, and Lauw 2021; Ren et al. 2022; Wang et al. 2022a; Zhang et al. 2023).

However, GAN-based and VAE-based models have some model issues (Wang et al. 2023). The adversarial training in GAN-based models is typically unstable, while VAE-based models suffer from the trade-off between tractability and representation ability. Most recently, some studies have begun utilizing Large Language Models (LLMs) for generative recommendation (Du et al. 2024; Zhang et al. 2024; Ji et al. 2024). However, these approaches still have a long way to go in terms of recommendation efficiency and throughput.

2.2 Diffusion Recommendation

Diffusion models have emerged as a new paradigm of generative models, and have shown high-quality generation capabilities (Croitoru et al. 2023; Popov et al. 2021), in both conditional (Chao et al. 2021; Ho, Jain, and Abbeel 2020; Liu et al. 2023b; Rombach et al. 2022) and unconditional generation (Austin et al. 2021). Recent studies explore their integration into recommendation.

CODIGEM (Walker et al. 2022) is the first collaborative filtering approach based on DDPM, which generates robust collaborative signals and latent representations. DiffRec (Wang et al. 2023) uses diffusion models to capture user interactions, reducing noise and incorporating temporal information for better accuracy. DiffuRec (Li, Sun, and Li 2023) leverages diffusion models for sequential recommendation, representing items as distributions to capture diverse user preferences. DiffuASR (Liu et al. 2023a) addresses data sparsity and long-tail user problems with a diffusion-based pseudo sequence generation framework, enhancing the quality of generated interactions for better model training. DreamRec (Yang et al. 2024) redefines sequential recommendation using a guided diffusion model, generating oracle items from historical interactions for precise user preference depiction. However, these studies ignore learning multiple distributions for each user, resulting in sub-optimal recommendation performance.

3 Preliminaries

In this section, we give a brief introduction to diffusion models, exemplified by Denoising Diffusion Probabilistic Models (DDPM) (Ho, Jain, and Abbeel 2020). DDPM can be divided into two main processes: 1) *Forward Process* that gradually adds noise to the data over several steps, and 2) *Reverse Process* that reconstructs the original data from the noisy version.

In the forward process, we start with a data point $\mathbf{x}_0 \in \mathbb{R}^{1\times d}$ drawn from the data distribution $q(\mathbf{x}_0)$ with dimension d. We then add Gaussian noise in small increments across T timesteps, resulting in a sequence of latent variables $\{\mathbf{x}_t\}_{t=1}^T$ ($\mathbf{x}_t \in \mathbb{R}^{1\times d}$). Mathematically, the forward diffusion process can be described as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

where β_t is a noise schedule parameter that controls the amount of noise added at each step. In DDPM, a reparameterization trick is used to obtain the closed form of output: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, where $\alpha_t = 1-\beta_t, \bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

The reverse process aims to denoise $\mathbf{x}_T$ back to the original data point $\mathbf{x}_0$ by learning the reverse conditional distributions $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. This process can be formulated as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)), \tag{2}$$

where μ_θ and Σ_θ are neural networks parameterized by θ that predict the mean and variance, respectively.

Training DDPM involves optimizing the parameters θ to minimize the difference between the true reverse process and the learned reverse process, which is achieved by maximizing the log-likelihood of the data or equivalently minimizing a variational bound on the negative log-likelihood:

$$\mathcal{L}(\theta) = \sum_{t=1}^T D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) + \mathbb{C}, \tag{3}$$

where $D_{KL}(\cdot\|\cdot)$ is the Kullback-Leibler (KL) divergence, and $\mathbb{C}$ is a constant and is independent of the model parameters θ. The posterior distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ can be solved using Bayes' theorem as:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I}), \tag{4}$$

where

$$\tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t, \tag{5}$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{6}$$

To solve $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, DDPM (Ho, Jain, and Abbeel 2020) sets $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \tilde{\beta}_t\mathbf{I}$ to match Eq. (4), and reparameterizes $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ by predicting $\boldsymbol{\epsilon}$ from $\mathbf{x}_t$. However, DiffRec (Wang et al. 2023) highlights that predicting $\mathbf{x}_0$ rather than $\boldsymbol{\epsilon}$ aligns more closely with the objective of ranking items in recommendation. Thus, they factorize $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ as:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}f_\theta(\mathbf{x}_t, t) + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t, \tag{7}$$

where f_θ is a function approximating $\mathbf{x}_0$ given $\mathbf{x}_t$ and t. In this case, Eq. (3) is transformed into:

$$\mathcal{L}(\theta) = \sum_{t=1}^T \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2}\left(\frac{\bar{\alpha}_{t-1}}{1-\bar{\alpha}_{t-1}} - \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}\right)\|\mathbf{x}_0 - f_\theta(\mathbf{x}_t, t)\|^2\right] + \mathbb{C}. \tag{8}$$

4 Methods

In this section, we first present the overview of the proposed GCDR model (see Figure 2), and then delve into the details of its approximator f_θ, followed by the training and inference procedure of GCDR.

4.1 Overview

Formally, we follow the standard setting where there exist a set of users $\mathcal{U}$, a set of items $\mathcal{I}$, a set of categories $\mathcal{C}$, and the interaction records $\mathcal{H}$ between $\mathcal{U}$ and $\mathcal{I}$. Each item $i \in \mathcal{I}$ is tagged with a subset of categories $C_i \subseteq \mathcal{C}$. The objective of the recommendation is to generate a set of items $\mathcal{R}_u \subseteq \mathcal{I}$ (typically the top-k items ranked by a recommendation model) for each user u.

GCDR learns multiple user distributions to sample user embedding for item ranking. Specifically, the user distributions are characterized as the distribution of the historically interacted items. As shown in Figure 2(a), this characterization conditions on the long-term interest encoded by user-specific embedding, and is guided by the short-term interest derived from recent interaction sequences, as well as the user category preference encoded by the category distribution in historically interacted items. Then, GCDR follows the DDPM framework, and leverages a forward process and a reverse process for model training and model inference, respectively (see Figures 2(b) and 2(c)). Note that in addition to the diffusion loss, GCDR is simultaneously optimized by the user loss and the recommendation loss, for better alignment with the objectives of the recommendation task.

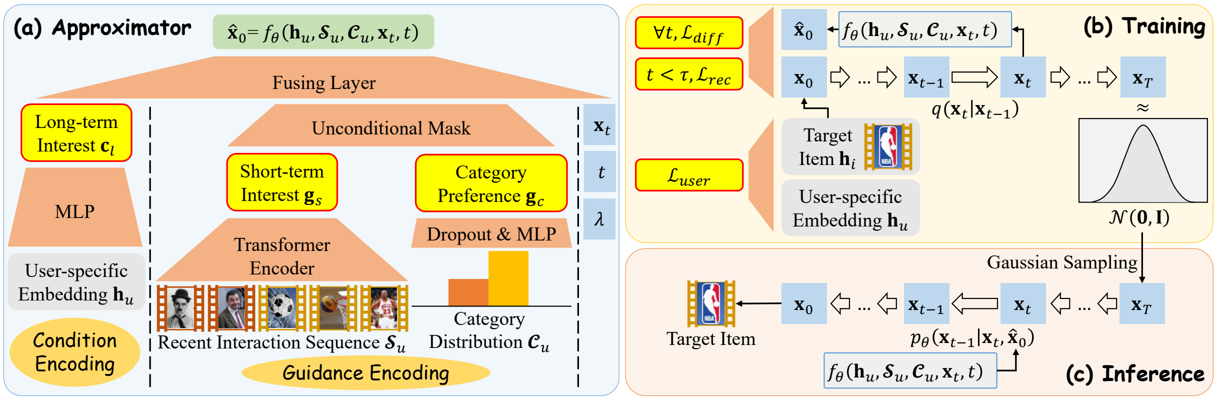


Figure 2: The architecture of GCDR. Figure 2(a) shows the details of the approximator, which encodes the long-term interest, the short-term interest and the category preference of a user for target item reconstruction. Figure 2(b) illustrates the training phase of GCDR, which is optimized via the user loss, the recommendation loss, and the diffusion loss. Figure 2(c) depicts the inference phase of GCDR, reconstructing the target item from a sampled Gaussian noise.

4.2 Approximator

The objective of GCDR's approximator is to predict $\mathbf{x}_0$ for diffusion. However, applying the approximator f_θ directly from Eq. (7) cannot facilitate personalized and guided diffusion. Therefore, as shown in Figure 2(a), given user u, GCDR employs an approximator formulated as:

$$\hat{\mathbf{x}}_0 = f_\theta(\mathbf{h}_u, \boldsymbol{\mathcal{S}}_u, \boldsymbol{\mathcal{C}}_u, \mathbf{x}_t, t). \quad (9)$$

where $\mathbf{h}_u$ represents user-specific embedding, $\mathcal{S}_u$ denotes the user's recent interaction sequence, and $\boldsymbol{\mathcal{C}}_u$ corresponds to the category distribution of the user's historical interactions. GCDR integrates these terms into the approximator through *Condition Encoding* and *Guidance Encoding*, and then outputs $\hat{\mathbf{x}}_0$ by the *Fusing Layer*.

Condition Encoding. The purpose of condition encoding is to enable inter-user distribution learning for personalized diffusion generation, providing the diffusion process with user-specific information and ensuring the overall consistency of distributions for each user. In this regard, GCDR randomly initializes a user-specific embedding $\mathbf{h}_u \in \mathbb{R}^{1\times d}$ for user u to distinguish between different users. This embedding is further encoded into the user's long-term interests $\mathbf{c}_l \in \mathbb{R}^{1\times d}$ (u is omitted for simplicity) via an MLP (Multi-Layer Perceptron) as follows:

$$\mathbf{c}_l = \text{MLP}(\mathbf{h}_u). \quad (10)$$

Note that in Section 4.3, we design a user loss to ensure that the user-specific embedding $\mathbf{h}_u$ captures the historical interactions of user u, which guarantees that $\mathbf{c}_l$ accurately represents the user's long-term interests.

Guidance Encoding. Unlike condition encoding, guidance encoding is designed for intra-user distribution learning, which is used to guide the model towards the appropriate sub-distribution given a user's specific distribution. There are two kinds of guidance in GCDR, namely *Short-term Interest* and *Category preference*.

The short-term interests of users are captured based on their recent interaction sequences. More precisely, given user u, we arrange her/his historical interactions in chronological order, and take the last M items to form the recent interaction sequence $\boldsymbol{\mathcal{S}}_u = \{\mathbf{s}_u^1, \mathbf{s}_u^2, \ldots, \mathbf{s}_u^M\}$, where $\mathbf{s}_u^m \in \mathbb{R}^{1\times d}$ denotes the embedding of the m-th item in the sequence. Then, we utilize a Transformer encoder to encode $\boldsymbol{\mathcal{S}}_u$ and obtain the short-term interest $\mathbf{g}_s$ (u is omitted for simplicity) as:

$$\mathbf{g}_s = \text{Transformer}(\boldsymbol{\mathcal{S}}_u). \quad (11)$$

The category preferences of users are calculated based on the categories of historically interacted items. Note that the categories of the target item are not included here to ensure that GCDR relies only on the users' historical interactions to infer their next interactions. Specifically, given user u and the corresponding historical interaction sequence $\mathcal{I}_u \subseteq \mathcal{I}$, the category distribution of the user's historically interacted items, denoted as $\boldsymbol{\mathcal{C}}_u$, is defined as a $|\mathcal{C}|$-dimensional vector:

$$\boldsymbol{\mathcal{C}}_u = \{p_u^{(c_1)}, p_u^{(c_2)}, \ldots, p_u^{(c_{|\mathcal{C}|})}\}, \quad (12)$$

$$p_u^{(c_k)} = \frac{f_u^{(c_k)}}{\sum_{c' \in \mathcal{C}} f_u^{(c')}}, \quad (13)$$

$$f_u^{(c)} = \sum_{i \in \mathcal{I}_u} \sum_{c' \in C_i} \mathcal{X}(c = c'), \quad (14)$$

where $\mathcal{X}(\cdot)$ is the indicator function, and $p_u^{(c_k)}$ is the frequency of category c_k appearing in the historical interactions. The category preference $\mathbf{g}_c$ (u is omitted for simplicity) is then derived as:

$$\mathbf{g}_c = \text{MLP}(\text{Dropout}(\boldsymbol{\mathcal{C}}_u)), \quad (15)$$

where the Dropout layer is employed to randomly deactivate certain dimensions within the category distribution. This technique prevents GCDR from overly relying on frequently occurring categories, allowing it to capture the user distributions across different categories, thereby enabling the learning of disentangled preferences.

According to ADM (Dhariwal and Nichol 2021), guided diffusion requires the assistance of an unconditional version. To achieve this, we adopt a classifier-free guidance scheme (Ho and Salimans 2021). Specifically, we randomly replace the guidance $\mathbf{g}_s$ and $\mathbf{g}_c$ with dummy tokens with probability u_s and u_c, respectively. Formally, we set two random variable $m_s \sim \text{Bernoulli}(u_s)$ and $m_c \sim \text{Bernoulli}(u_c)$, and use them for *Unconditional Mask*:

$$\mathbf{g}'_s = (1 - m_s) \cdot \mathbf{g}_s + m_s \cdot \boldsymbol{\psi}_s, \tag{16}$$

$$\mathbf{g}'_c = (1 - m_c) \cdot \mathbf{g}_c + m_c \cdot \boldsymbol{\psi}_c, \tag{17}$$

where $\boldsymbol{\psi}_s, \boldsymbol{\psi}_c \in \mathbb{R}^{1\times d}$ are the dummy tokens for unconditional diffusion.

Fusing Layer. The condition encoding $\mathbf{c}_l$ and guidance encodings $\mathbf{g}'_s$ and $\mathbf{g}'_c$ are used in the fusing layer to estimate $\hat{\mathbf{x}}_0$, formally defined as follows:

$$\hat{\mathbf{x}}_0 = \text{MLP}(\mathbf{c}_l \oplus \mathbf{g}'_s \oplus \mathbf{g}'_c \oplus (\delta \cdot \mathbf{x}_t) \oplus \mathbf{e}_t), \tag{18}$$

where $\oplus$ denotes vector concatenation, and $\mathbf{e}_t \in \mathbb{R}^{1\times d}$ is the timestep embedding at time t. δ is a hyper-parameter that controls the influence of $\mathbf{x}_t$. A smaller δ implies greater personalization, while a larger δ signifies more randomness. In this manner, the approximator of GCDR can predict $\hat{\mathbf{x}}_0$ by not only being conditioned on a specific user u, but also taking into account the user's long-term and short-term interests as well as category preferences, thereby enhancing the diffusion generation effect.

4.3 Model Training

As illustrated in Figure 2(b), the overall training process of GCDR adheres to that of DDPM in Section 3. The data point to be diffused here is the embedding of the target item (*i.e.*, $\mathbf{x}_0 = \mathbf{h}_i$, where $\mathbf{h}_i$ is randomly initialized) since the user distributions are captured as the distribution of the interacted items. To train GCDR, we incorporate the *User Loss* and the *Recommendation Loss* in addition to the *Diffusion Loss*.

The user loss is employed to train the user-specific embedding $\mathbf{h}_u$, enabling it to encode the user's historical interaction sequence. More precisely, given user u and item i she/he has interacted with, we first randomly sample N negative samples. We then maximize the similarity between u and i while minimizing the similarity between u and the negative samples. Formally, the user loss is defined as:

$$\mathcal{L}_{user} = -\log \sigma(\mathbf{h}_u^\top \mathbf{h}_i) - \sum_{n=1}^{N} \log \sigma(-\mathbf{h}_u^\top \tilde{\mathbf{h}}_n). \tag{19}$$

where $\mathbf{h}_i$ represents the embedding of item i, $\tilde{\mathbf{h}}_n$ denotes the embedding of the n-th negative sample, and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. We use the cosine similarity to measure the similarity between users and items as many recommendation models do.

The recommendation loss is used to optimize the approximator so that its output $\hat{\mathbf{x}}_0 = f_\theta(\mathbf{h}_u, \boldsymbol{\mathcal{S}}_u, \boldsymbol{\mathcal{C}}_u, \mathbf{x}_t, t)$ can approximate the embedding of the target item $\mathbf{x}_0$ within the recommendation context. Similar to user loss, the recommendation loss is defined based on cosine similarity and negative sampling:

$$\mathcal{L}_{rec} = -\log \sigma(\hat{\mathbf{x}}_0^\top \mathbf{x}_0) - \sum_{n=1}^{N} \log \sigma(-\hat{\mathbf{x}}_0^\top \tilde{\mathbf{h}}_n). \tag{20}$$

Datasets	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{I}\rvert$	# Interactions	$\lvert\mathcal{C}\rvert$
MovieLens	6,028	3,422	575,056	18
KuaiRec	7,176	10,722	1,514,281	31
Yelp	146,949	125,879	2,397,886	100
Books	712,802	1,874,004	9,838,152	100

Table 1: The statistics of datasets.

Note that we compute $\mathcal{L}_{rec}$ only when $t \leq \tau$, and the intuition is that when t is relatively large, $\mathbf{x}_t$ progressively approaches Gaussian noise, making it challenging for the computed $\hat{\mathbf{x}}_0$ to recover the original $\mathbf{x}_0$. Consequently, calculating the recommendation loss in such situations might lead to performance degradation.

The diffusion loss of GCDR follows the form of Eq. (8) and is used to optimize the approximator so that its output $\hat{\mathbf{x}}_0$ can be effectively utilized for diffusion generation during the inference phase. The diffusion loss is formally defined as:

$$\mathcal{L}_{diff} = \frac{1}{2}\left(\frac{\bar{\alpha}_{t-1}}{1-\bar{\alpha}_{t-1}} - \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}\right)\|\mathbf{x}_0 - \hat{\mathbf{x}}_0\|^2. \tag{21}$$

It is important to note that although both the recommendation loss and the diffusion loss are computed based on $\mathbf{x}_0$ and $\hat{\mathbf{x}}_0$, they have different focuses. The former is designed to adapt to the recommendation task, while the latter aims to train the diffusion model.

Finally, the global training loss is the combination of the above three losses:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_{user} \cdot \mathcal{L}_{user} + \lambda_{diff} \cdot \mathcal{L}_{diff}, \tag{22}$$

where λ_{user} and λ_{diff} are hyper-parameters to control the influences of $\mathcal{L}_{user}$ and $\mathcal{L}_{diff}$, respectively.

4.4 Model Inference

During the inference phase, given user u, GCDR reconstructs the target item's embedding by iteratively denoising from Gaussian noise (*i.e.*, $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$) based on the trained approximator. Combining Eqs. (2) and (7), the reverse transition formula from time step t to $t-1$ is as follows:

$$\mathbf{x}_{t-1} = \left(\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\mathbf{x}}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t\right) + \sqrt{\tilde{\beta}_t}\boldsymbol{\epsilon}. \tag{23}$$

To better leverage the multiple user distributions learned by GCDR, we made two adjustments to the approximator during the inference phase. The first adjustment involves removing the *Unconditional Mask* step as:

$$\hat{\mathbf{x}}_0^\phi = \text{MLP}(\mathbf{c}_l \oplus \mathbf{g}_s \oplus \mathbf{g}_c \oplus (\delta \cdot \mathbf{x}_t) \oplus \mathbf{e}_t). \tag{24}$$

The second adjustment aims to achieve intentional and guided embedding sampling, *i.e.*, to generate target item embedding based on the user's distribution w.r.t. a specific category. Specifically, we select the top-L categories from the user's category distribution to construct one-hot category distribution vectors $\{\boldsymbol{\mathcal{O}}_u^1, \ldots, \boldsymbol{\mathcal{O}}_u^L\}$. These vectors replace $\boldsymbol{\mathcal{C}}_u$ in Eq. (15) to obtain $\{\mathbf{g}_c^1, \ldots, \mathbf{g}_c^L\}$, leading to category-specific estimates $\hat{\mathbf{x}}_0^l$:

$$\hat{\mathbf{x}}_0^l = \text{MLP}(\mathbf{c}_l \oplus \mathbf{g}_s \oplus \mathbf{g}_c^l \oplus (\delta \cdot \mathbf{x}_t) \oplus \mathbf{e}_t). \tag{25}$$

Dataset	MovieLens			KuaiRec			Yelp			Books		
Metric	Recall	HR	NDCG	Recall	HR	NDCG	Recall	HR	NDCG	Recall	HR	NDCG
MF	0.0750	0.5037	0.0745	0.0518	0.9235	0.1312	0.0445	0.0991	0.0227	0.0939	0.1585	0.0467
NCF	0.0741	0.5032	0.0773	0.0501	0.9440	0.1360	0.0417	0.0951	0.0210	0.0860	0.1499	0.0441
NGCF	0.0788	0.5231	0.0793	0.0499	0.9320	0.1358	0.0248	0.0677	0.0115	0.0348	0.0674	0.0152
LightGCN	0.0795	0.5212	0.0801	0.0507	0.9348	0.1362	0.0881	0.1887	0.0463	0.1165	0.1966	0.0602
GRU4REC	0.0622	0.4621	0.0576	0.0434	0.8724	0.1351	0.0311	0.0762	0.0166	0.0449	0.0835	0.0241
SASRec	0.0762	0.5019	0.0803	0.0589	<u>0.9539</u>	0.1388	0.0394	0.0926	0.0207	0.0718	0.1307	0.0378
STOSA	0.1131	0.5509	0.0911	0.0506	<u>0.9405</u>	0.1399	0.1013	0.2357	0.0534	0.0551	0.1040	0.0287
HCUR	0.0622	0.4621	0.0576	<u>0.0617</u>	0.9369	0.1552	0.0359	0.0854	0.0179	0.0797	0.1399	0.0413
CBox4CR	0.0714	0.4910	0.0633	0.0372	0.8356	0.1198	0.0303	0.0741	0.0150	0.0958	0.1582	0.0477
LCD-UC	0.0817	0.5228	0.0819	0.0557	0.9461	<u>0.1695</u>	0.0959	0.2092	0.0498	0.1104	0.1900	0.0569
DiffRec	0.0987	<u>0.5568</u>	0.0932	0.0594	0.9448	0.1508	0.0452	0.0997	0.0232	0.0936	0.1583	0.0490
DiffuRec	<u>0.1265</u>	0.5299	<u>0.0976</u>	0.0563	0.9497	0.1671	<u>0.1371</u>	<u>0.2772</u>	<u>0.0711</u>	<u>0.1273</u>	<u>0.2097</u>	<u>0.0665</u>
DreamRec	0.0645	0.5011	0.0711	0.0220	0.7123	0.0820	0.0387	0.0901	0.0198	0.0867	0.1383	0.0436
GCDR	**0.1319**	**0.6025**	**0.1076**	**0.0733**	**0.9681**	**0.1778**	**0.1552**	**0.3086**	**0.0809**	**0.1363**	**0.2244**	**0.0711**

Table 2: Overall Performance. The best result is in bold, and the runner-up is underlined. Note that GCDR achieves statistically significant improvements under t-test compared to other baselines with $p < 0.01$ on MovieLens, $p < 0.05$ on KuaiRec, $p < 0.0001$ on Yelp, and $p < 0.00001$ on Books.

By substituting $\{\hat{\mathbf{x}}_0^\phi, \hat{\mathbf{x}}_0^1, \ldots, \hat{\mathbf{x}}_0^L\}$ into $\hat{\mathbf{x}}_0$ in Eq. (23), we can reconstruct $\mathbf{X}_0 = \{\mathbf{x}_0^\phi, \mathbf{x}_0^1, \ldots, \mathbf{x}_0^L\}$. Then, the similarity between each candidate item i and the user u is calculated as follows:

$$s(u,i) = \max_{\mathbf{x}_0 \in \mathbf{X}_0} \mathbf{x}_0^\top \mathbf{h}_i. \tag{26}$$

The recommendation list $\mathcal{R}_u$ for the user comprises the top-K items from the candidate set that have the highest similarity scores with the user.

Note that the overall complexity for each training epoch in GCDR is $O((M^2 + N) \cdot d + M \cdot d^2)$, while the time complexity of the inference phase is $O(L \cdot T \cdot (M^2 \cdot d + M \cdot d^2))$. Appendix 1 presents the details of time complexity analysis, as well as the training and inference algorithm.

5 Experiments

In this section, we conduct extensive experiments to evaluate our proposed GCDR.

5.1 Experimental Settings

Datasets. The performance of GCDR is evaluated using four real-world datasets of varying scales and scenarios, including MovieLens (Harper and Konstan 2015), KuaiRec (Gao et al. 2022), Yelp (Wang et al. 2023) and Books (Wang et al. 2023). The statistics of these datasets are summarized in Table 1, and the details can be found in Appendix 2.1.

Baseline Methods. We evaluate GCDR with thirteen state-of-the-art methods, categorized into the following four groups: 1) *Non-sequential Methods*, including MF (Koren, Bell, and Volinsky 2009), NCF (He et al. 2017), NGCF (Wang et al. 2019), LightGCN (He et al. 2020), 2) *Sequential Methods*, including GRU4REC (Hidasi et al. 2015), SASRec (Kang and McAuley 2018), STOSA (Kang and McAuley 2018), 3) *Box-embedding Methods*, including HCUR (Zhang et al. 2021), CBox4CR (Liang et al. 2023), LCD-UC (Wu et al. 2024), and 4) *Diffusion Methods*, including DiffRec (Wang et al. 2023), DiffuRec (Li, Sun, and Li 2023), DreamRec (Yang et al. 2024). The details of the baseline methods can be found in Appendix 2.2, while the parameter settings of all models are listed in Appendix 2.3.

Metrics. We adopt three popular evaluation measures namely **Recall@**k, HitRate@k (**HR@K**), and Normalized Discounted Cumulative Gain (**NDCG@k**) for performance comparison (Shani and Gunawardana 2011). Note that we report $k = 20$ and similar results are observed when $k = 3$, $k = 5$, and $k = 10$.

5.2 Overall Performance

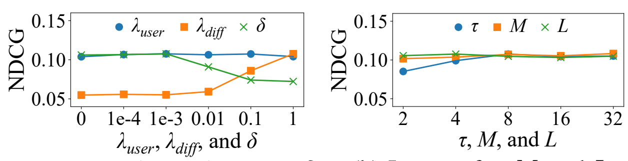
We present the performance of GCDR and the baseline methods in Table 2. We can observed that: 1) GCDR achieves the best performance across all metrics on all datasets, owing to the usage of conditional guided diffusion for multiple user distributions learning. 2) GCDR outperforms sequential methods and box-embedding methods, indicating that diffusion models have a superior capability in capturing sequential information and learning distributions. 3) GCDR excels over other diffusion methods because it captures both inter-user and intra-user distributions.

5.3 Ablation Study

In this subsection, we evaluate the effectiveness of different components of GCDR with three groups of variants: 1) ***w/o*** $\mathbf{c}_l$, ***w/o*** $\mathbf{g}_s$, and ***w/o*** $\mathbf{g}_c$ where the long-term interest $\mathbf{c}_l$, the short-term interest $\mathbf{g}_s$, and the category preference $\mathbf{g}_c$ are not included in the fusing layer, respectively; 2) ***w/o*** $\mathcal{L}_{user}$, ***w/o*** $\mathcal{L}_{rec}$ and ***w/o*** $\mathcal{L}_{diff}$ where the user loss $\mathcal{L}_{user}$, the recommendation loss $\mathcal{L}_{rec}$, and the diffusion loss $\mathcal{L}_{diff}$ are not employed in the training phase, respectively. 3) ***w/o*** $\mathbf{x}_0^l$ which only uses $\mathbf{x}_0^\phi$ to measure the similarity between user u and item i by $s(u,i) = (\mathbf{x}_0^\phi)^\top \mathbf{h}_i$, *i.e.*, the variant employing only

Dataset Metric	MovieLens Recall	MovieLens NDCG	KuaiRec Recall	KuaiRec NDCG
w/o $\mathbf{c}_l$	0.1209	0.0988	0.0731	**0.1778**
w/o $\mathbf{g}_s$	0.1019	0.0980	0.0729	0.1773
w/o $\mathbf{g}_c$	0.1252	0.1033	0.0726	0.1768
w/o $\mathcal{L}_{user}$	0.1242	0.1019	0.0561	0.1470
w/o $\mathcal{L}_{rec}$	0.0049	0.0067	0.0023	0.0133
w/o $\mathcal{L}_{diff}$	0.0581	0.0531	0.0568	0.1488
w/o $\mathbf{x}_0^l$	0.1265	0.1049	0.0718	0.1751
GCDR	**0.1319**	**0.1076**	**0.0733**	**0.1778**

Table 3: Ablation Studies. The best result is in bold, and the runner-up is underlined.



(a) Impact of λ_{user}, λ_{diff}, and δ (b) Impact of τ, M, and L

Figure 3: The experimental results of parameter sensitivity.

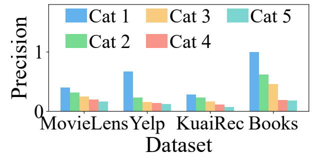


Figure 4: The effectiveness of category guidance.

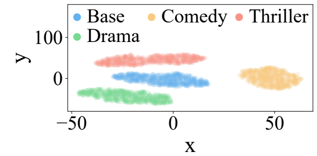


Figure 5: Distributions visualization of user $u = 463$.

a single user distribution without category-specific distributions.

We report the experimental results on MovieLens and KuaiRec with Recall and NDCG in Table 3. It is observed that: 1) Both the conditional encoding $\mathbf{c}_l$ and the guidance encodings $\mathbf{g}_s$ as well as $\mathbf{g}_c$ enhance the performance of GCDR, demonstrating the advantage of learning disentangled and personalized distribution. 2) The three losses $\mathcal{L}_{user}$, $\mathcal{L}_{rec}$, and $\mathcal{L}_{diff}$ all play crucial roles in model training. In particular, $\mathcal{L}_{rec}$ is especially important as it directly relates to the recommendation task. 3) Utilizing $\mathbf{x}_0^l$ for category-specific guidance can improve model performance, thereby validating the effectiveness of the intentional and guided embedding sampling in GCDR.

5.4 Parameter Sensitivity

We analyze the sensitivity of hyper-parameters in GCDR w.r.t. the NDCG metric on MovieLens. According to Figure 3(a), GCDR is not very sensitive to the user loss coefficient λ_{user}, with an optimal setting of $\lambda_{user} = 0.001$. However, GCDR requires a relatively large diffusion loss coefficient ($\lambda_{diff} = 1$) and a small $\mathbf{x}_t$ coefficient ($\delta = 0.001$). Figure 3(b) illustrates that the threshold τ for calculating the recommendation loss should not be too small, as excessively high values may also have negative effects due to the noise $\mathbf{x}_t$. Setting $\tau = 8$ is found to be optimal, Additionally, a longer length M of recent interaction sequences tends to yield better results, with $M = 8$ striking a good balance between performance and computational overhead. When providing category-specific guidances, using too many categories may introduce noise. Thus, setting $L \in \{3, 4, 5\}$ is most appropriate. The influence of other hyper-parameters such as d, T, N, u_s, and u_c can be found in Appendix 2.4.

5.5 Effectiveness of Category Guidance

We further investigate how category guidance takes effect in GCDR. Specifically, we analyze the proportion of items that share the same category as the guiding category with maximum similarity. Figure 4 presents the experimental results, where "Cat i" represents the i-th frequently interacted category of each user. As observed, the proportion of category-consistent items is higher when using frequently interacted categories as guidances. This result aligns with intuition, as less frequently interacted categories imply greater uncertainty. It is noteworthy that although we do not use the knowledge about target item's categories during training (which is reserved for future work), GCDR still achieves a satisfactory precision performance, especially on Yelp and Books. This outcome validates the effectiveness of intentional embedding sampling.

5.6 Distributions Visualization

We visually illustrate the multiple user distributions learned by GCDR. Specifically, we randomly sample a user from MovieLens. Then, for each $\hat{\mathbf{x}}_0 \in \{\hat{\mathbf{x}}_0^{\phi}, \hat{\mathbf{x}}_0^1, \ldots, \hat{\mathbf{x}}_0^L\}$, we draw 1,000 samples from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and use Eq. (23) to generate 1,000 user embeddings through the reverse denoising steps. These embeddings are reduced in dimensionality using t-SNE (Van der Maaten and Hinton 2008) and plotted. Figure 5 displays the distributions where "Base" refers to the distribution generated using $\hat{\mathbf{x}}_0^{\phi}$. The user embeddings generated with the same guidance are closely clustered in the space. Additionally, the distributions of different categories are clearly separated, while the "Base" distribution is in the middle as it represents an average distribution. This result aligns with our motivation and demonstrates the effectiveness of GCDR in learning multiple user distributions. We provide another user visualization in Appendix 2.5.

6 Conclusion

In this paper, we presented GCDR, a guided conditional recommendation model to represent each user with multiple distributions. The long-term interest was encoded using a user-specific embedding, while the short-term interest was incorporated via depicting the recent interaction sequence as guidance. We also leveraged the category preference to enable category-aware embedding sampling. Then, GCDR was trained by three losses, aiming to align the diffusion model with the recommendation task. The comprehensive experimental results empirically showed that GCDR can generate high-quality and effective user embeddings to improve recommendation performances.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 62372264) and Kuaishou Inc. Chaokun Wang is the corresponding author.

References

Altenburger, K. M.; and Ho, D. E. 2019. Is Yelp actually cleaning up the restaurant industry? A re-analysis on the relative usefulness of consumer reviews. In *WWW*, 2543–2550.

Austin, J.; Johnson, D. D.; Ho, J.; Tarlow, D.; and Van Den Berg, R. 2021. Structured denoising diffusion models in discrete state-spaces. *NeurIPS*, 34: 17981–17993.

Chang, J.; Gao, C.; Zheng, Y.; Hui, Y.; Niu, Y.; Song, Y.; Jin, D.; and Li, Y. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the 44th International ACM SIGIR*, 378–387.

Chao, C.-H.; Sun, W.-F.; Cheng, B.-W.; Lo, Y.-C.; Chang, C.-C.; Liu, Y.-L.; Chang, Y.-L.; Chen, C.-P.; and Lee, C.-Y. 2021. Denoising Likelihood Score Matching for Conditional Score-based Data Generation. In *ICLR*.

Chen, H.; Wang, Z.; Huang, F.; Huang, X.; Xu, Y.; Lin, Y.; He, P.; and Li, Z. 2022. Generative adversarial framework for cold-start item recommendation. In *Proceedings of the 45th International ACM SIGIR*, 2565–2571.

Croitoru, F.-A.; Hondru, V.; Ionescu, R. T.; and Shah, M. 2023. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9): 10850–10869.

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

Du, Y.; Luo, D.; Yan, R.; Wang, X.; Liu, H.; Zhu, H.; Song, Y.; and Zhang, J. 2024. Enhancing job recommendation through llm-based generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 8363–8371.

Gao, C.; Li, S.; Lei, W.; Chen, J.; Li, B.; Jiang, P.; He, X.; Mao, J.; and Chua, T.-S. 2022. KuaiRec: A Fully-Observed Dataset and Insights for Evaluating Recommender Systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, 540–550.

Gao, M.; Zhang, J.; Yu, J.; Li, J.; Wen, J.; and Xiong, Q. 2021. Recommender systems based on generative adversarial networks: A problem-driven perspective. *Information Sciences*, 546: 1166–1185.

Gu, T.; Wang, C.; Wu, C.; Lou, Y.; Xu, J.; Wang, C.; Xu, K.; Ye, C.; and Song, Y. 2022. HybridGNN: Learning Hybrid Representation for Recommendation in Multiplex Heterogeneous Networks. In *ICDE*. IEEE.

Gu, Y.; Ding, Z.; Wang, S.; and Yin, D. 2020. Hierarchical user profiling for e-commerce recommender systems. In *Proceedings of the 13th International Conference on WSDM*, 223–231.

Guo, G.; Zhou, H.; Chen, B.; Liu, Z.; Xu, X.; Chen, X.; Dong, Z.; and He, X. 2020. IPGAN: Generating informative item pairs by adversarial sampling. *IEEE transactions on neural networks and learning systems*, 33(2): 694–706.

Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *TIIS*, 5(4): 1–19.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR*, 639–648.

He, X.; He, Z.; Du, X.; and Chua, T.-S. 2018. Adversarial personalized ranking for recommendation. In *The 41st International ACM SIGIR*, 355–364.

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.

He, X.; Zhang, H.; Kan, M.-Y.; and Chua, T.-S. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR*, 549–558.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C. P.; Glorot, X.; Botvinick, M. M.; Mohamed, S.; and Lerchner, A. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3.

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. 2021. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

Ji, J.; Li, Z.; Xu, S.; Hua, W.; Ge, Y.; Tan, J.; and Zhang, Y. 2024. Genrec: Large language model for generative recommendation. In *European Conference on Information Retrieval*, 494–502. Springer.

Jin, B.; Lian, D.; Liu, Z.; Liu, Q.; Ma, J.; Xie, X.; and Chen, E. 2020. Sampling-decomposable generative adversarial recommender. *Advances in Neural Information Processing Systems*, 33: 22629–22639.

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

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.

Li, S.; Kawale, J.; and Fu, Y. 2015. Deep collaborative filtering via marginalized denoising auto-encoder. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, 811–820.

Li, X.; and She, J. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 305–314.

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.

Liang, T.; Zhang, Y.; Di, Q.; Xia, C.; Li, Y.; and Yin, Y. 2023. Contrastive Box Embedding for Collaborative Reasoning. In *Proceedings of the 46th International ACM SIGIR*, 38–47.

Liu, Q.; Yan, F.; Zhao, X.; Du, Z.; Guo, H.; Tang, R.; and Tian, F. 2023a. Diffusion augmentation for sequential recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 1576–1586.

Liu, S.; Sun, F.; Ge, Y.; Pei, C.; and Zhang, Y. 2021. Variation control and evaluation for generative slate recommendations. In *Proceedings of the Web Conference 2021*, 436–448.

Liu, X.; Park, D. H.; Azadi, S.; Zhang, G.; Chopikyan, A.; Hu, Y.; Shi, H.; Rohrbach, A.; and Darrell, T. 2023b. More control for free! image synthesis with semantic diffusion guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 289–299.

Liu, Z.; Wang, C.; Feng, H.; Wu, L.; and Yang, L. 2022. Knowledge distillation based contextual relevance matching for e-commerce product search. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 63–76.

Ma, J.; Zhou, C.; Cui, P.; Yang, H.; and Zhu, W. 2019. Learning disentangled representations for recommendation. *Advances in neural information processing systems*, 32.

Popov, V.; Vovk, I.; Gogoryan, V.; Sadekova, T.; and Kudinov, M. 2021. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*, 8599–8608. PMLR.

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*, 89–98.

Ren, Z.; Tian, Z.; Li, D.; Ren, P.; Yang, L.; Xin, X.; Liang, H.; de Rijke, M.; and Chen, Z. 2022. Variational reasoning about user preferences for conversational recommendation. In *proceedings of the 45th international ACM SIGIR*, 165–175.

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.

Salah, A.; Tran, T. B.; and Lauw, H. 2021. Towards source-aligned variational models for cross-domain recommendation. In *Proceedings of the 15th ACM Conference on Recommender Systems*, 176–186.

Sedhain, S.; Menon, A. K.; Sanner, S.; and Xie, L. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web*, 111–112.

Shani, G.; and Gunawardana, A. 2011. Evaluating recommendation systems. *Recommender systems handbook*, 257–297.

Shenbin, I.; Alekseev, A.; Tutubalina, E.; Malykh, V.; and Nikolenko, S. I. 2020. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In *Proceedings of the 13th international conference on web search and data mining*, 528–536.

Truong, Q.-T.; Salah, A.; and Lauw, H. W. 2021. Bilateral variational autoencoder for collaborative filtering. In *Proceedings of the 14th ACM international conference on web search and data mining*, 292–300.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

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

Wang, J.; Yu, L.; Zhang, W.; Gong, Y.; Xu, Y.; Wang, B.; Zhang, P.; and Zhang, D. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR*, 515–524.

Wang, W.; Lin, X.; Feng, F.; He, X.; Lin, M.; and Chua, T.-S. 2022a. Causal representation learning for out-of-distribution recommendation. In *Proceedings of the ACM Web Conference 2022*, 3562–3571.

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*, 832–841.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 165–174.

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

Wu, C.; Shi, S.; Wang, C.; Liu, Z.; Peng, W.; Wu, W.; Kong, D.; Li, H.; and Gai, K. 2024. Enhancing Recommendation Accuracy and Diversity with Box Embedding: A Universal Framework. In *Proceedings of the ACM on Web Conference 2024*, WWW '24, 3756–3766. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701719.

Wu, C.; Wang, C.; Xu, J.; Fang, Z.; Gu, T.; Wang, C.; Song, Y.; Zheng, K.; Wang, X.; and Zhou, G. 2023a. Instant Representation Learning for Recommendation over Large Dynamic Graphs. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 81–94. IEEE.

Wu, C.; Wang, C.; Xu, J.; Liu, Z.; Zheng, K.; Wang, X.; Song, Y.; and Gai, K. 2023b. Graph Contrastive Learning with Generative Adversarial Network. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2721–2730.

Wu, Q.; Liu, Y.; Miao, C.; Zhao, B.; Zhao, Y.; and Guan, L. 2019. PD-GAN: Adversarial learning for personalized diversity-promoting recommendation. In *IJCAI*, volume 19, 3870–3876.

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.

Xu, J.; Wang, C.; Wu, C.; Song, Y.; Zheng, K.; Wang, X.; Wang, C.; Zhou, G.; and Gai, K. 2023. Multi-behavior Self-supervised Learning for Recommendation. In *Proceedings of the 46th International ACM SIGIR*, SIGIR '23, 496–505. New York, NY, USA: Association for Computing Machinery. ISBN 9781450394086.

Xu, L.; Lian, J.; Zhao, W. X.; Gong, M.; Shou, L.; Jiang, D.; Xie, X.; and Wen, J.-R. 2022. Negative sampling for contrastive representation learning: A review. *arXiv preprint arXiv:2206.00212*.

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.

Yu, X.; Zhang, X.; Cao, Y.; and Xia, M. 2019. VAEGAN: A collaborative filtering framework based on adversarial variational autoencoders. In *IJCAI*, volume 19, 4206–4212.

Zhang, A.; Chen, Y.; Sheng, L.; Wang, X.; and Chua, T.-S. 2024. On generative agents in recommendation. In *Proceedings of the 47th International ACM SIGIR*, 1807–1817.

Zhang, S.; Liu, H.; Zhang, A.; Hu, Y.; Zhang, C.; Li, Y.; Zhu, T.; He, S.; and Ou, W. 2021. Learning user representations with hypercuboids for recommender systems. In *Proceedings of the 14th ACM international conference on web search and data mining*, 716–724.

Zhang, S.; Yao, L.; and Xu, X. 2017. Autosvd++ an efficient hybrid collaborative filtering model via contractive auto-encoders. In *Proceedings of the 40th International ACM SIGIR*, 957–960.

Zhang, Y.; Zhang, Y.; Yan, D.; Deng, S.; and Yang, Y. 2023. Revisiting graph-based recommender systems from the perspective of variational auto-encoder. *ACM Transactions on Information Systems*, 41(3): 1–28.