Computing and Informatics, Vol. 45, 2026, 55–81, doi: 10.31577/cai_2026_1_55

CDGAN: COLLABORATIVE DIFFUSION GENERATIVE ADVERSARIAL NETWORKS FOR RECOMMENDATION SYSTEMS

Jinzhong Li*

School of Electronic and Information Engineering
Jinggangshan University
Ji'an 343009, China
&
Key Laboratory of Embedded System and Service Computing (Tongji University)
Ministry of Education
Shanghai 201804, China
e-mail: `lijinzhong@jgsu.edu.cn`

Junqi Liu, Zhiqiang Huang

School of Software Engineering
Jiangxi University of Science and Technology
Nanchang 330013, China
e-mail: `{6720231505, huangzhiqiang}@mail.jxust.edu.cn`

Abstract. Deep generative models are widely used in recommendation systems because of their ability to deal with uncertainty by learning inherent data distribution. Among deep generative models, Generative Adversarial Networks (GAN) perform well in recommendation tasks. However, existing Collaborative Filtering (CF) recommendation algorithms based on GAN generally have problems such as mode collapse and training instability, which are further aggravated by sparse and noisy recommendation data. To solve these problems, a Collaborative Diffusion Generative Adversarial Networks (CDGAN) framework for recommendation systems is proposed in this paper. Specifically, CDGAN framework is mainly composed of three parts: feature encoder, diffusion generator, and self-attention discriminator.

---

* Corresponding author

The feature encoder extracts the rating information and side information to obtain potential feature vectors to alleviate the data sparsity problem. The diffusion generator simulates the complex nonlinear mode of the user-item interaction matrix through forward diffusion and reverse denoising to reconstruct the user-item interaction matrix to alleviate the problem of data noise. The self-attention discriminator obtains the user's specific behaviors and preferences through the self-attention mechanism to improve the discriminator's discriminating ability. Furthermore, a corresponding CDGAN recommendation algorithm is designed based on our proposed CDGAN framework. Comprehensive experiments are conducted on three real-world recommendation datasets. The experimental results indicate that, when compared with multiple representative recommendation models, the proposed CDGAN model achieves superior performance in the evaluation metrics Precision and Recall on all datasets, thereby proving its effectiveness.

Keywords: Recommendation systems, collaborative filtering, generative adversarial networks, diffusion models, autoencoder, self-attention mechanism, rating information, side information

1 INTRODUCTION

With the advent of the big data era, the application of neural networks in recommendation systems has increasingly underscored its significance. Amidst the explosive growth of data, users confront the dilemma of information overload, struggling to promptly access the information they need. Recommendation systems alleviate this issue by mining the characteristic information and correlations between users and items, thereby assisting users in uncovering content that piques their interests from vast amounts of data. At present, a variety of recommendation algorithms have been proposed for the recommendation systems, such as content-based recommendation [1, 2], CF-based recommendation [3, 4, 5, 6, 7, 8], and hybrid recommendation [9, 10]. Additionally, many recommendation algorithms are applied in fields such as POI recommendation [11], trip recommendation [12], location recommendation [13], and group recommendation [14]. Collaborative Filtering (CF), as a core technology in recommendation systems, it is widely applied through analyzing and comparing users' historical behavior data to identify items that users may be interested in. Among them, CF models based on deep neural networks have attracted significant attention in both industry and academia due to their powerful feature extraction and learning capabilities. These models can effectively capture user preferences from their historical interactions while safeguarding user privacy, without requiring excessive auxiliary and sensitive information.

In recent years, Generative Adversarial Networks (GAN) [15] and its variants have been increasingly applied in CF, such as IRGAN [4] and CFGAN [5] which leverage generative adversarial training to learn more accurate user preferences and item features, thereby enhancing the performance of recommendations. However, these methods are plagued by issues such as mode collapse and training instability. Moreover, with the exponential growth of data volume, recommendation systems are confronted with challenges posed by data noise [16] and data sparsity [17] which further exacerbate these problems. Data noise may lead to the deviation of user behavior patterns and reduce the effectiveness of recommendations. To address the issue of data noise, recently, researchers have started using deep generative models [18] to learn the underlying structure and distribution of data, thereby reducing the impact of data noise on recommendation results. On the other hand, data sparsity stems from the fact that users typically interact with only a small subset of products, making it difficult to fully reflect user interests. To tackle the problem of data sparsity, researchers have begun to focus on the utilization of side information. Side information includes additional clues such as user profiles and product descriptions (in this study, we use user profiles as side information, including the user's name, gender, occupation, etc.), which provides the recommendation systems with more dimensional information. Currently, some studies have started to extract nonlinear features from side information to assist model training, thereby more accurately predicting user interests and preferences [19, 20]. However, as the side information usually has high-dimensional characteristics [21], direct use may lead to difficulties in training.

To address the above problems, we integrate Diffusion Models (DM) [22] and AutoEncoders (AE) [23] into GAN and propose a model of Collaborative Diffusion Generative Adversarial Networks for recommendation systems, termed as CDGAN. Compared with traditional recommendation models, CDGAN can learn more effective representations from the data and effectively reduce the impact of noise on the recommendation results. Specifically, we treat collaborative filtering as a click-through rate prediction problem and generate a user-item interaction matrix through GAN. The feature encoder uses AE to extract features and reduce dimensions of user behavior data and user side information, aiming to convert high-dimensional side information into low-dimensional latent feature vectors. This method not only solves the problem of high-dimensional data but also improves the ability of GAN to mine sparse data. The diffusion generator integrates the DM as the generator, reconstructing the user-item rating matrix through forward diffusion and reverse denoising processes, thereby removing data noise and improving the stability of GAN. The self-attention discriminator uses the self-attention mechanism to learn local features and global structures in the data, enhancing its ability to distinguish between real and generated data, thereby indirectly promoting the generator to produce predicted ratings close to the real ratings. The main contributions of this study can be summarized as follows.

- A novel GAN-based recommendation framework called CDGAN is proposed and a corresponding CDGAN recommendation algorithm is designed. The CDGAN framework mainly consists of three parts: feature encoder, diffusion generator, and self-attention discriminator.

- CDGAN uses the feature encoder to enhance the original rating matrix by extracting latent feature vectors from both the rating information and the side information, avoiding data high-dimensional issues, and alleviating the problem of data sparsity, thereby achieving better performance.
- CDGAN utilizes the DM as the generator to reconstruct the user-item interaction matrix and integrates the objective function of the DM into the loss function of the generator. This method effectively enhances the noise-handling and training stability of the GAN, thereby strengthening the model's predictive performance.
- CDGAN integrates the self-attention mechanism into the discriminator to enhance its discriminative ability by capturing specific user behaviors and preferences.
- We conduct extensive experiments on three real-world datasets to validate the effectiveness of the CDGAN model compared to four representative baseline models. Additionally, we perform multiple ablation experiments and the experimental results demonstrate the effectiveness of these components within the CDGAN model.

The remaining parts of this paper are organized as follows. Section 2 presents the relevant work. In Section 3, we introduce our proposed CDGAN framework and the corresponding recommendation algorithm in detail. Section 4 presents a series of experiments to verify the validity of our model. Finally, Section 5 concludes the study and discusses future work.

2 RELATED WORK

2.1 Generative Adversarial Networks Based Recommendation Systems

Generative Adversarial Networks (GAN) [15] have attracted immense attention since their birth. As a type of deep generative model, GAN generates synthetic data similar to real data through adversarial training between the generator and the discriminator, GAN and its variants have been widely applied in recommendation systems. IRGAN [4] pioneeringly introduced GAN into the field of recommendation systems. Through adversarial training of a combination of a generative retrieval model and a discriminative retrieval model, and by applying the reinforcement learning based on policy gradient algorithm, it has successfully solved the difficult problem that discrete data is hard to optimize by gradient descent. Consequently, IRGAN can generate samples that are of high quality and close to real data distribution, bringing a brand-new training idea to the field of recommendation systems. GraphGAN [6] applies graph representation learning to capture complex relationships between users and items, enhancing the performance of generative and discriminative models through adversarial training. Additionally, it introduces a graph softmax function and an online generation strategy to optimize computational efficiency and sample quality. However, these methods still have some issues. Since the samples generated by the generator are discrete data, a single discrete item index is generated each time. Furthermore, as training progresses, the generator may produce items that are identical to the real data, confusing the discriminator and ultimately reducing recommendation performance. To solve this problem, Chae et al. [5] proposed CFGAN and innovatively adopted the vectorized adversarial training method to enable the generator to generate continuous-valued purchase vectors, avoiding the problem of the discriminator's performance degradation caused by label contradictions. Through this training method, the CFGAN model can learn the continuity and diversity of user preferences and significantly improve the recommendation performance. Due to the innovative ideas of IRGAN and CFGAN and their far-reaching influence in the recommendation systems, they are considered to be the milestone models of GAN-based recommendation systems, and the subsequent newly proposed recommendation models usually choose these two models for comparison. Ren et al. [24] incorporated the self-attention mechanism into GAN and proposed MRGAN. The generator captures user preferences through multi-head parallel processing to generate personalized recommendations, while the discriminator utilizes a bidirectional structure to provide feedback that enhances the model's effectiveness and interpretability. However, these methods are confronted with issues of mode collapse and training instability, primarily due to the generator's difficulty in capturing all the features of the data distribution when the dataset is sparse or contains excessive noise, leading to a lack of diversity in the generated samples. Consequently, the discriminator may encounter a local optimum and feedback failure, exacerbating the instability of GAN training.

Therefore, we propose a novel model of CDGAN to address these issues in this paper. CDGAN enhances the original rating matrix by extracting latent nonlinear features from user rating information and side information through a feature encoder to alleviate the problem of data sparsity. In addition, compared to traditional GAN-based CF models, CDGAN integrates the DM as the generator and uses the self-attention mechanism to assist in the training of the discriminator, thereby effectively mitigating the problem of data noise.

2.2 Side Information in Recommendation Systems

Various methods have been developed to integrate side information [25] such as user-item information into CF models to improve recommendation performance, such as latent factor models, representation learning models, and deep learning models, etc. [26]. In this section, we only explore methods that use deep learning models to obtain the nonlinear features of side information to improve the performance of recommendation systems. Wu et al. [27] proposed a Hybrid Conditional Variational Autoencoder (HCVAE), which leverages an AE network to extract user features and utilizes an optimization objective combined with side information of both users and items to construct more expressive latent representations. Wei et al. [28] introduced a recommendation model named GRCN based on graph convolutional neural network, which improves the structure of the interaction graph adaptively by utilizing the rich content of items and the historical behaviors of users, thereby enhancing the performance of recommendations. Wen et al. [29] presented a GAN framework for personalized recommendation called PRGAN, which takes user and item subsets as conditional information, uses a memory module to store the embedding representations of users and items, and a classification module that uses a convolutional neural network to capture local patterns to distinguish between real and generated data. Zhou and Chen [7] proposed a Variational Collaborative Generative Adversarial Network (VCGAN), which extracts low-dimensional latent vectors of user information and side information through an AE and uses a Variational Autoencoder (VAE) [30] to learn a reasonable representation of the data.

Different from the aforementioned works, in this paper, we use two feature encoders to extract nonlinear features from user rating information and side information to alleviate data sparsity and improve the performance of recommendations.

2.3 Diffusion Models Based Recommendation Systems

Diffusion Models (DM) [22], as a type of deep generative model, perturb the input data by adding random noise through a forward diffusion process, and then iteratively learn to gradually restore the original data during the reverse denoising process. Since user-item interaction data in recommendation systems often contains noise, DM can learn and recover these damaged interactions, thereby improving the accuracy and personalization of recommendations. Recently, some studies have attempted to apply DM for recommendation systems. Walker et al. [8] proposed CODIGEM, the first CF model based on the denoising diffusion probabilistic model (DDPM), which generates powerful collaborative signals and robust latent representations by simulating the complex nonlinear patterns of user-item interaction data. Wang et al. [31] proposed a novel diffusion recommendation model called DiffRec, which retains personalized information of user interactions by controlling the noise scale and inference steps in the forward process, thereby extending the traditional DM to reduce resource costs for large-scale item prediction. Li et al. [32] proposed a DM-based sequential recommendation model called DiffuRec, which represents items as distributions rather than fixed vectors, thus more flexibly capturing users' multiple interests and the multifaceted characteristics of items, and enhancing recommendation performance through target item guidance. Wu et al. [33] proposed Diff4Rec, which combines DM with curriculum learning strategies to enhance sequential recommendation systems and generates diversified data augmentation samples by simulating the corruption and recovery process of user-item interactions in the latent space. Wang et al. [34] proposed Diff-MSR, which applies DM to multi-scenario recommendations under cold-start conditions. By designing novel variance scheduling and classifiers, the recommendation performance for cold-start scenarios is enhanced. Liu et al. [35] proposed a DM-based data augmentation method DiffASR, combining the generative capabilities of DM with the predictive capabilities of sequence U-Nets, to generate high-quality pseudo-sequence data.

Given the many advantages of applying DM to recommendation systems, we adopt the DM as a key component of our model and integrate it into the generator of GAN. By simulating the complex nonlinear patterns of the user-item interaction matrix through the forward diffusion and reverse denoising of the DM, the user-item interaction matrix is reconstructed to address the issue of data noise.

3 PROPOSED CDGAN FRAMEWORK AND ALGORITHM

In this study, we propose a novel recommendation system framework, namely the Collaborative Diffusion Generative Adversarial Networks (CDGAN), and design the corresponding CDGAN recommendation algorithm. Figure 1 illustrates the CDGAN recommendation framework, and Algorithm 1 outlines the procedure of the CDGAN recommendation algorithm. CDGAN consists of three parts: the feature encoder, the diffusion generator, and the self-attention discriminator. First, two feature encoders are used to extract features from user ratings and side information to obtain low-dimensional latent vectors. Then, the latent vectors are concatenated with the rating matrix to form a new rating matrix as the input for the diffusion generator, which helps to overcome the problem of data sparsity. The diffusion generator uses the DM to learn the generative process of user interactions through denoising, generating predicted user-item interaction vectors that better conform to user preferences, thereby effectively overcoming the problem of data noise. The self-attention discriminator uses the self-attention mechanism to deeply understand the complex features and preference patterns in the real data, thus more effectively distinguishing between real and generated samples, and then indirectly promoting the generator to produce data samples that are closer to the real data through its feedback mechanism.

3.1 Feature Encoder

To mitigate the issue of data sparsity and improve recommendation performance, we utilize Autoencoder (AE) as Feature Encoder (FE) to separately extract features from the user-item rating matrix and user-side information. Assuming user index $u \in \{1, \ldots, m\}$, item index $i \in \{1, \ldots, n\}$, and user feature index $f \in \{1, \ldots, d\}$, the Rating Feature Encoder (RFE) takes the user-item rating matrix $UI \in \mathbb{R}^{m \times n}$ as input, while the Side Information Feature Encoder (SIFE) takes the user-feature matrix $UF \in \mathbb{R}^{m \times d}$ as input. Initially, we binarize the user rating information: When user u interacts with item i, the rating value is set to 1; otherwise, it is set to 0. For user-side information, we encode it in one-hot encoding mode and map the user features into vectors. We employ RFE and SIFE to extract latent features and reconstruct the input matrices through Equations (1) and (2):

$$L_{UI} \sim f_\theta(UI), \quad UI' \sim g_\phi(L_{UI}), \tag{1}$$

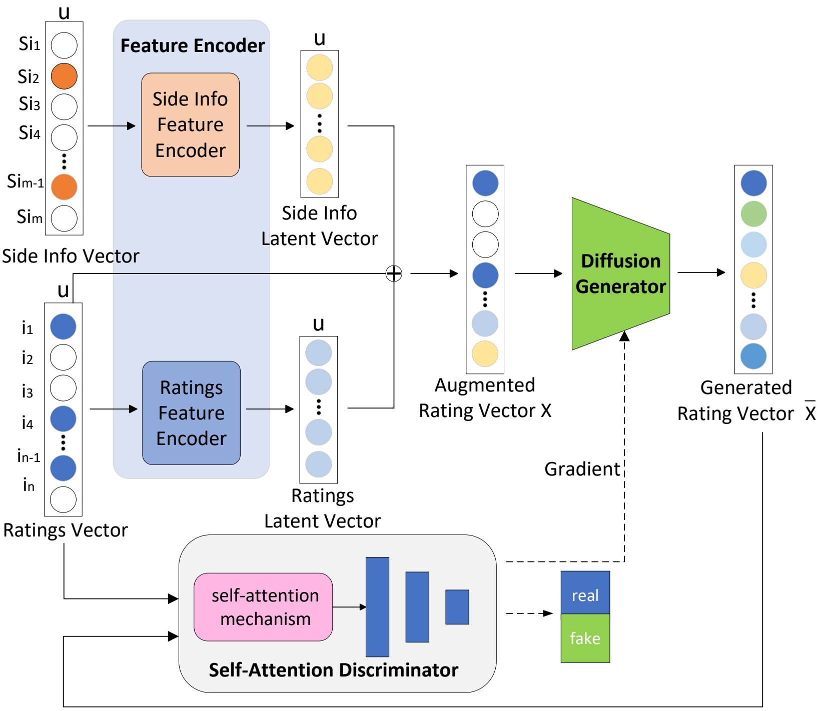

Figure 1. Architecture of the CDGAN framework

$$L_{UF} \sim f_\theta(UF), \quad UF' \sim g_\phi(L_{UF}), \tag{2}$$

where $f_\theta(\cdot)$ and $g_\phi(\cdot)$ correspond to the encoding and decoding networks parameterized by θ and ϕ, respectively. L_{UI} and L_{UF} correspond to the nonlinear low-dimensional latent vectors of the user-item rating matrix UI and the user-feature matrix UF, respectively. UI' and UF' represent the reconstructed user-item rating matrix and the user-feature matrix, respectively. We use the mean squared error (MSE) as the loss for the feature encoder to minimize the reconstruction error, as shown in Equations (3) and (4):

$$\mathcal{L}(UI, UI') = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (UI_{ij} - UI'_{ij})^2, \tag{3}$$

$$\mathcal{L}(UF, UF') = \frac{1}{m \times d} \sum_{i=1}^{m} \sum_{j=1}^{d} (UF_{ij} - UF'_{ij})^2. \tag{4}$$

The notation UI_{ij} represents the rating value of user i for item j and UF_{ij} represents the one-hot encoding value of the j^{th} feature of user i. UI'_{ij} refers to the transformed rating value of user i for item j as reconstructed by the decoder network, and UF'_{ij} is the transformed one-hot encoding value of the j^{th} feature of user i as reconstructed by the decoder network. After the training of the RFE and SIFE is completed, and the extracted latent vectors $L_{UI} \in \mathbb{R}^{m\times p}$ and $L_{UF} \in \mathbb{R}^{m\times q}$ are obtained. Here, $p \ll n$ and $q \ll d$. Finally, we concatenate L_{UI} and L_{UF} with the user rating matrix UI to get the enhanced rating matrix $X \in \mathbb{R}^{m\times r}$ as the input for the diffusion generator, where $r = (n + p + q)$.

3.2 Diffusion Generator

Currently, DM is widely used in the field of image generation due to its powerful generative capability and denoising characteristics [36, 37, 38]. To overcome data noise and issues such as instability and mode collapse that traditional GAN may encounter during the training process, we introduce DM as the generator of GAN within CDGAN to generate high-quality recommendation data. DM can provide a stable initialization state, making it easier for GAN to converge during the training process. In addition, the denoising characteristics of DM also help to eliminate noise and redundant information in the data generated by GAN, thereby improving data quality. Figure 2 illustrates the structure of the diffusion generator.

The diffusion generator consists of two processes: the forward diffusion process and the reverse denoising process. The forward diffusion process gradually introduces noise into the original data samples, causing the perturbation to approach Gaussian noise. Conversely, the reverse denoising process restores the original samples by predicting and subtracting the added noise step by step. Next, we will provide a detailed introduction to these two processes.

3.2.1 Forward Diffusion Process

Given an input data sample $X = x_0 \sim q(x_0)$, the forward diffusion process $q(x_{1:T}|x_0)$ constructs a Markov chain of latent variables by gradually adding Gaussian noise over T time steps. Specifically, this process generates a noisy sequence $x_1, \ldots, x_T$ which can be represented by Equations (5) and (6):

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}), \tag{5}$$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{6}$$

where $\beta_t \in (0, 1)$ is utilized to control the scale of Gaussian noise added at each step t. $\mathbf{I}$ is the identity matrix. By using the reparameterization trick [22] and the additivity of two independent Gaussian noises [39], we can directly obtain x_t from x_0, which is formalized in Equation (7):

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{7}$$

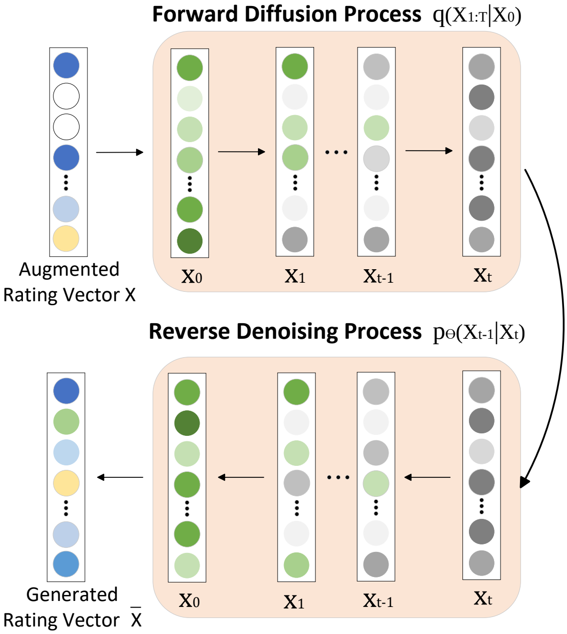

Figure 2. The structure of the diffusion generator

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, $\alpha_t = 1 - \beta_t$, and then we can reparameterize $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

3.2.2 Reverse Denoising Process

The purpose of the reverse denoising process $p_\theta(x_{0:T})$ is to denoise the noise-added x_t and recover the original interaction vector x_0. Specifically, given the current denoised representation x_t, the next representation x_{t-1} is obtained after one step of reversal. The reverse denoising process commences with $p(x_T) = \mathcal{N}(x_T; \mathbf{0}, \mathbf{I})$ and progressively executes Gaussian transitions. It is parameterized into Equations (8) and (9) by utilizing a Markov chain:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta\left(x_{t-1} | x_t\right), \tag{8}$$

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right), \tag{9}$$

where the mean $\mu_\theta(x_t, t)$ and variance $\Sigma_\theta(x_t, t)$ are Gaussian parameters output by a deep neural network (DNN) with parameters θ.

3.2.3 The Learning Process of Diffusion Generator

Since we designed the DM as the generator for GAN, to ensure that the generated data follows a custom distribution and to avoid outliers, we design the objective function of the diffusion generator as two parts, namely the loss of DM based on KL divergence $\mathcal{L}_{DM}$ and the log-likelihood loss of the original generator $\mathcal{L}_G$, which is formalized as Equation (10):

$$J_G = \alpha \cdot \mathcal{L}_{DM} + \beta \cdot \mathcal{L}_G, \tag{10}$$

where the $\mathcal{L}_{DM}$ is given by Equation (12), and $\mathcal{L}_G$ is given by Equation (14); α and β are hyperparameters that balance $\mathcal{L}_{DM}$ and $\mathcal{L}_G$.

The Loss of DM Based on KL Divergence. Given the original distribution $x_0 \sim X$, the purpose of training is to generate a data distribution that is very close to the actual data distribution through the reverse denoising process $p_\theta(x_{0:T})$. Parameters θ can be learned by optimizing the variational lower bound of negative log-likelihood, which can be further expressed as the combination of KL divergence and entropy terms [22], and formalized as Equation (11).

$$\mathbb{E}\left[-\log p_\theta(x_0)\right] \leq \mathbb{E}_{(x_{0:T})\sim q}\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right], \tag{11}$$

where $\mathbb{E}_{(x_{0:T})\sim q}\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right]$ in Equation (11) represents the loss function of DM, termed as $\mathcal{L}_{DM}$, which can be rewritten as Equation (12).

$$\begin{aligned}\mathcal{L}_{DM} = \mathbb{E}_q\Bigg[&D_{KL}\left(q(x_T|x_0)\|p_\theta(x_T)\right)\\ &+\sum_{t=2}^{T} D_{KL}\left(q(x_{t-1}|x_t, x_0)\|p_\theta(x_{t-1}|x_t)\right)\\ &-\log p_\theta(x_0|x_1)\Bigg],\end{aligned} \tag{12}$$

where D_{KL} represents the KL divergence between the true denoising distribution $q(\cdot)$ and the parameterized denoising distribution $p_\theta(\cdot)$.

The Log-Likelihood Loss of the Original Generator. The training process of GAN is a minimax game between the generator G and the discriminator D. The generator produces a predicted user-item interaction matrix to deceive the discriminator, which learns to judge whether the generator's output conforms to the true posterior distribution based on the real user-item interaction matrix. The generator optimizes its parameters based on the feedback results from the discriminator. The overall objective function of GAN can be represented by

Equation (13):

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{Z\sim P_{\text{data}}}\left[\log D(Z)\right] + \mathbb{E}_{\bar{X}\sim P_\theta}\left[\log\left(1 - D(G(X))\right)\right]. \tag{13}$$

Here, Z is the real data from the data distribution P_{data}, while $\bar{X}$ is the generated data from the model distribution of $G(X)$, and X is the input data for the generator. When the parameters of the discriminator are fixed, the training objective of the generator is to minimize the probability that its generated data is classified as fake data by the discriminator, and its loss function can be expressed as Equation (14):

$$\mathcal{L}_G = \mathbb{E}_{\bar{X}\sim P_\theta}\left[\log\left(1 - D(G(X))\right)\right] = \frac{1}{m}\sum_{i=1}^{m}\left(\log\left(1 - D(G(x_i))\right)\right), \tag{14}$$

where x_i represents the user-item interaction vector for user i generated from the distribution by the generator G.

3.3 Self-Attention Discriminator

The task of the discriminator is primarily to distinguish whether the input samples come from the real data samples or fake samples generated by the generator, and to provide key feedback signals to the generator. Therefore, in existing GAN-based CF frameworks, the discriminator is often regarded as a binary classifier. Although the neural network structure designed in this way can effectively perform basic discrimination tasks, the potential of the discriminator is far from being fully exploited in the whole GAN framework. In fact, the discriminator plays a dual role during the training process: on the one hand, it needs to learn how to distinguish between the real samples and generated samples. On the other hand, its gradient information is also used to update the parameters of the generator, thereby helping the generator to produce more realistic data.

Therefore, we expand the discriminator into a neural network capable of extracting explicit user preferences through a self-attention mechanism. This design allows the discriminator to not only complete the discrimination task but also to share the pressure of the generator. By obtaining feature information of user preferences, it guides the learning process of the generator, thereby simplifying the task of the generator. The self-attention mechanism maps queries and a set of key-value pairs to outputs through an attention function, where the queries, keys, values, and outputs are all vectors. The output is calculated as a weighted sum of these values, where the weights assigned to each value are computed by the compatibility function of the query and the corresponding key [40]. In this paper, we employ scaled dot-product attention to extend the discriminator, which can be defined as Equation (15).

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{Q\cdot K^T}{\sqrt{d_k}}\right)V, \tag{15}$$

where Q, K, and V stand for Query, Key, and Value, respectively. In this study, Q, K, and V all originate from linear transformations of the discriminator's input. By calculating the dot product of Q and K, the similarity between the two vectors Q and K can be obtained, and the weight of V can be determined. When the dimension of the key matrix K is large, the scaling factor $\sqrt{d_k}$ is used to control and prevent this problem, that is, the inner product is too large to make the gradient smaller, which leads to difficulty in training.

For the training of the discriminator, the loss function is designed to minimize the classification error, this is, to maximize the probability of correctly classifying the real samples and fake samples. According to Equation (13), when the parameters of the generator are fixed, its loss function is defined as Equation (16):

$$\begin{aligned}\mathcal{L}_D &= -\mathbb{E}_{Z\sim P_{\text{data}}}\left[\log D(Z)\right] - \mathbb{E}_{\bar{X}\sim P_\theta}\left[\log\left(1 - D(G(X))\right)\right], \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left[\log(D(z_i)) + \log\left(1 - D(G(x_i))\right)\right], \end{aligned} \tag{16}$$

where z_i represents the user-item interaction vector of user i from the real data distribution. The first term in $\mathcal{L}_D$ represents the loss of the real samples, and the second term represents the loss of the generated samples.

3.4 CDGAN Recommendation Algorithm

Based on the above-mentioned CDGAN recommendation framework, this section designs a specific CDGAN recommendation algorithm. We design the CDGAN recommendation algorithm into two stages:

Stage 1: Extracting latent feature vectors by using FE.

In this stage, UI and UF are encoded, and these matrices are reconstructed from the latent space to obtain latent feature vectors L_{UI} and L_{UF}. Then, L_{UI} and L_{UF} are concatenated with UI to form the enhanced rating matrix X.

Stage 2: Adversarial training generates predictive items.

In this stage, the enhanced rating matrix X is input into the GAN framework, and their respective parameters are continuously optimized through iteratively adversarial training of the diffusion generator and the self-attention discriminator. Ultimately, the diffusion generator can learn the complex distribution of user-item interactions and generate high-quality recommendation items.

Through the collaborative work of these two stages, the CDGAN algorithm can provide more accurate predictions of user preferences, thereby improving user experience and satisfaction. In addition, subsequent experiments further verify that it has significant advantages over other models in key metrics such as Precision and Recall, and it can generate more accurate and comprehensive recommended items.

The detailed workflow of the CDGAN recommendation algorithm is shown in Algorithm 1.

Algorithm 1 CDGAN Recommendation Algorithm

```
Input: user-item rating matrix $\boldsymbol{UI}$, user-feature matrix $\boldsymbol{UF}$, feature encoder FE, generator $G$ and discriminator $D$, learning rate $\mu_{FE}$, $\mu_G$ and $\mu_D$
Output: $G$'s parameter $\theta$, $D$'s parameter $\phi$
Initialize: FE's parameter $\eta$, $G$'s parameter $\theta$, $D$'s parameter $\phi$, FE-step, $G$-step, $D$-step
// Extracting latent feature vectors by using FE
for FE-step do
    Sample a minibatch from $\boldsymbol{UI}$ and $\boldsymbol{UF}$
    Encode the minibatch data into the latent space and decode it back, reconstructing $\boldsymbol{UI}$ and $\boldsymbol{UF}$ to obtain $\boldsymbol{UI'}$ and $\boldsymbol{UF'}$ by Equations (1) and (2)
    Calculate reconstruction loss $\mathcal{L}$ by Equations (3) and (4)
    Update FE by $\eta \leftarrow \eta - \mu_{FE} \cdot \nabla_{\eta}\mathcal{L}$
end for
Get latent vectors $L_{UI}$ and $L_{UF}$ by FE
Concat $L_{UI}$, $L_{UF}$ and $\boldsymbol{UI}$ to obtain the enhanced rating matrix $\boldsymbol{X}$.
// Adversarial training generates predictive Items
while not converged do
    // Training of the Diffusion Generator
    for $G$-step do
        Sample a minibatch from $\boldsymbol{X}$
        Generate fake rating vectors through forward diffusion and reverse denoising processes
        Calculate the loss $J_G$ of the Diffusion Generator by Equation (10)
        Update $G$ by $\theta \leftarrow \theta - \mu_G \cdot \nabla_{\theta} J_G$
    end for
    // Training of the Self-Attention Discriminator
    for $D$-step do
        Sample a minibatch from $\boldsymbol{X}$
        Generate fake rating vectors through forward diffusion and reverse denoising processes
        Calculate the loss $\mathcal{L}_D$ of the Self-Attention Discriminator by Equation (16)
        Update $D$ by $\phi \leftarrow \phi - \mu_D \cdot \nabla_{\phi}\mathcal{L}_D$
    end for
end while
return $\theta$
```

4 EXPERIMENTS AND RESULTS ANALYSIS

4.1 Datasets

The experiments were conducted on three public datasets: MovieLens 100K, MovieLens 1M[1], and Steam[2]. Table 1 summarizes the characteristics of these datasets. The MovieLens 100K and MovieLens 1M are movie rating datasets collected from the MovieLens website, while the Steam dataset is a collection of users' purchase records and game duration from the Steam gaming platform. The sparsity of the Steam dataset reaches as high as 99.89 %, even sparser than many large datasets, which can be used to demonstrate the ability of CDGAN to handle extremely sparse data. For each dataset, we randomly divide the user-item interaction data into two subsets: 80 % for training and the remaining 20 % for testing.

DATASET	Users	Items	Ratings	Sparsity
MovieLens 100K	943	1 682	100 000	93.69 %
MovieLens 1M	6 040	3 952	1 000 209	95.81 %
Steam	12 393	5 155	70 490	99.89 %

Table 1. Dataset statistics

4.2 Evaluation Metrics

We employ two common metrics used in the top-N recommendations, which are Precision@N and Recall@N, which respectively reflect the accuracy rate and recall rate of the recommendation systems, and their respective calculation methods are shown in Equations (17) and (18).

$$Precision@N = \frac{|relevant\_items \cap recommended\_items|}{N}, \tag{17}$$

$$Recall@N = \frac{|relevant\_items \cap recommended\_items|}{|relevant\_items|}, \tag{18}$$

where *relevant_items* is the list of related items for each user in the test data, *recommended_items* represents the list of top-N related items for each user in the generated data of the model, and N is the total number of recommended items, which is set to 5, 10, and 20, respectively, in these experiments.

4.3 Implementation Details

The experiments utilize an NVIDIA GeForce RTX 3080 Ti GPU with CUDA-accelerated PyTorch and an Intel Xeon Silver 4210R CPU for training.

---

[1] `https://grouplens.org/datasets/movielens`

[2] `https://www.kaggle.com/datasets/tamber/steam-video-games`

In the RFE and SIFE, the encoder network and decoder network are symmetrical, comprising two hidden layers with {1024, 128, 64} neurons. In the diffusion generator, the structures of the forward and reverse processes are also symmetrical, containing five hidden layers with {200, 500, 1 000} neurons. The PRelu activation function is used between layers, and the Tanh activation function is used in the last layer of the reverse process. The hyperparameter β_t is set to 0.0001, and the diffusion step T is set to 3. The learning rates μ_{FE}, μ_G, and μ_D of the feature encoder, generator, and discriminator are all set to 0.0001 initially. To facilitate better convergence during the training process, we employ a learning rate scheduler to dynamically adjust the learning rates of the generator and discriminator. Specifically, when the epoch reaches 30, 200, and 400, the learning rates of the generator and discriminator are reduced to 10 % of their original values, respectively. Additionally, we set the parameters α and β in Equation (10) to 1, respectively. In the self-attention discriminator, the number of hidden nodes in each hidden layer is {1 024, 128, 16, 1}. During the training process, our model is trained by using the Adam method [41] with a batch size of 64. Each dataset is trained for 500 epochs, the diffusion generator is updated 5 times and the self-attention discriminator is updated 2 times in each epoch.

4.4 Baseline Models and Comparative Results

We compare the CDGAN model with the following representative collaborative filtering baseline models to verify its performance.

- **IRGAN** [4] integrates generative and discriminative models through adversarial training, where the generative model is responsible for producing items related to users' queries, while the discriminative model is responsible for distinguishing whether these items are genuinely relevant or generated by the generative model.
- **CFGAN** [5] utilizes users' historical behavioral data as conditions and trains the model through vectorized generative adversarial methods to enhance the accuracy and diversity of recommendations. In CFGAN, the generator is responsible for generating recommended items, while the discriminator evaluates the relevance of the recommendations.
- **VCGAN** [7] uses AE to extract implicit feedback from user ratings and side information, and employs VAE as the generator to learn a reasonable representation of the data to predict user ratings for items, while the discriminator distinguishes between predicted ratings and actual ratings.
- **CODIGEM** [8] is the first collaborative filtering recommendation model based on the denoising diffusion probabilistic model (DDPM). It effectively simulates user-item interaction data through forward and reverse diffusion processes, captures complex nonlinear patterns, and generates strong collaborative signals and robust latent representations, thereby improving the generalization ability and recommendation performance of the model.

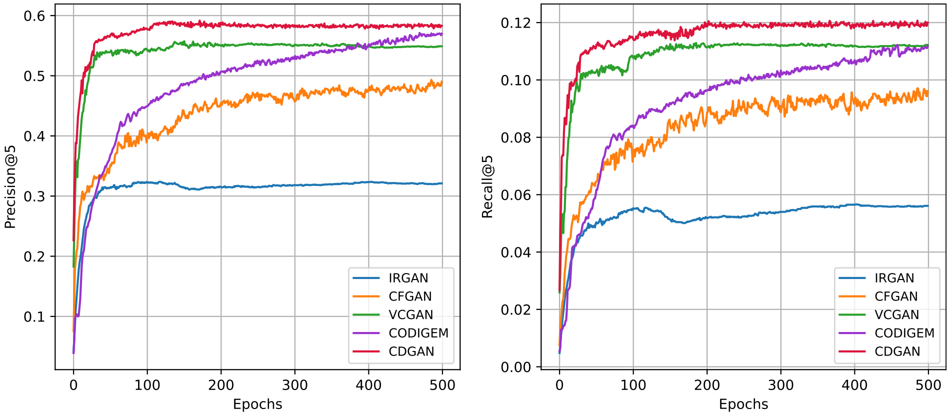

Figure 3. The learning curves on MovieLens 100K dataset

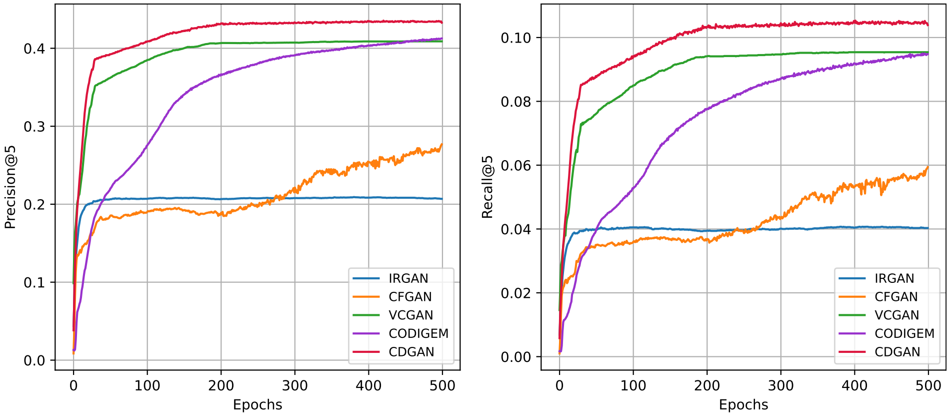

Figure 4. The learning curves on MovieLens 1M dataset

Based on the CDGAN model proposed in this paper and all the above-mentioned baseline models, experimental results are obtained on the three datasets of MovieLens 100K, MovieLens 1M, and Steam. To make the models more comparable, for each baseline model, we test various values of its hyperparameters and select the optimal parameters that provide them with the highest precision. Figures 3, 4, and 5 show the learning curves of CDGAN and these baseline models on three real-world datasets. It can be seen that CDGAN exhibits higher stability during the training process and achieves significant performance improvement. This is mainly due to the autoregressive training approach and the gradual denoising process adopted by DM, which avoids the problems of mode collapse and mode noise, thereby improving

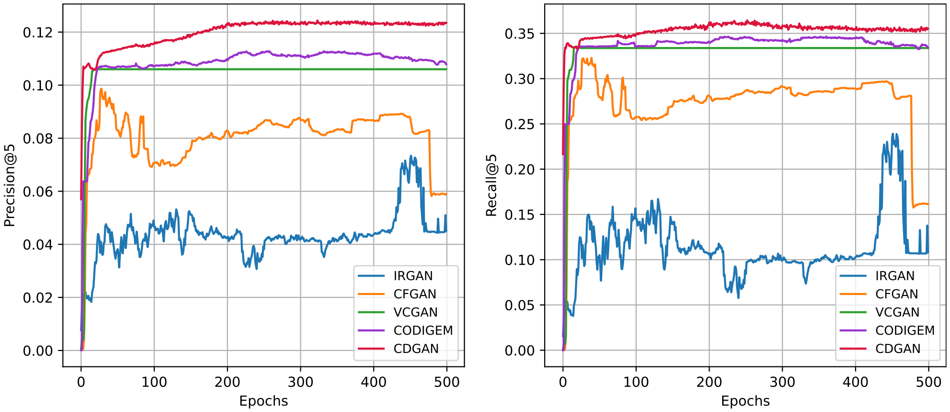

Figure 5. The learning curves on Steam dataset

the training stability and performance of the model. Table 2 presents the performance comparison results of CDGAN and other baseline models on three different datasets. The experimental results show that CDGAN achieves the best performance across all metrics on all datasets. Specifically, compared to the best baseline model, CDGAN achieves a 3.73 % improvement in Precision@5 and a 4.53 % improvement in Recall@5 on the MovieLens 100K dataset. On the MovieLens 1M dataset, CDGAN improves Precision@5 by 5.55 % and Recall@5 by 9.96 %. Furthermore, on the Steam dataset, CDGAN enhances Precision@5 by 10.28 % and Recall@5 by 4.49 %. Among all GAN-based baseline models, VCGAN and CDGAN exhibit the most similar performance. Unlike VCGAN, CDGAN employs DM as the generator and incorporates a self-attention mechanism in the discriminator. As can be seen from the table, CDGAN exhibits significant performance improvements compared with VCGAN. This is primarily because DM uses a gradual diffusion approach to generate samples, which helps the model capture the complex nonlinear characteristics of user behavior, thereby generating high-quality and diverse samples. The self-attention mechanism enhances the discriminator's ability to distinguish between real data and generated data by capturing the explicit preference features of users. Additionally, our model also outperforms another DM-based recommendation model, CODIGEM. This is primarily because CDGAN utilizes the feature encoder to extract and concatenate both rating and feature information, effectively mitigating the impact of data sparsity on model performance.

Furthermore, Table 2 also reveals a downward trend in Precision as the number of recommended items N increases across the three datasets. This is because when the denominator N increases, the growth speed of the numerator (the number of correctly recommended items) usually fails to keep up with the increase of N. Conversely, Recall exhibits an opposite trend, because as N increases, the numerator has a greater trend of increasing, while the denominator (the number of relevant items) remains the same. Notably, Precision on the MovieLens dataset is generally higher than Recall, while Precision on the Steam dataset is lower than Recall. This is because the Steam dataset is too sparse, that is, the number of relevant items per user is small. Consequently, for the same numerator, N may be greater than the number of relevant items, so the Precision is higher than the Recall.

Models	**Precision@5**	**Precision@10**	**Precision@20**	**Recall@5**	**Recall@10**	**Recall@20**
			MovieLens 100K			
IRGAN	0.3237	0.3017	0.2515	0.0554	0.0967	0.1459
CFGAN	0.4902	0.4246	0.3720	0.0961	0.1592	0.2562
VCGAN	0.5573	0.4845	0.4101	0.1125	0.1795	0.2799
CODIGEM	0.5704	0.4967	0.4236	0.1104	0.1810	0.2872
CDGAN	**0.5917**	**0.5131**	**0.4319**	**0.1176**	**0.1941**	**0.3010**
Improvement	3.73 %	3.30 %	1.96 %	4.53 %	7.24 %	4.81 %
			MovieLens 1M			
IRGAN	0.2092	0.1812	0.1537	0.0407	0.0685	0.1149
CFGAN	0.2768	0.2381	0.2002	0.0594	0.0984	0.1564
VCGAN	0.4090	0.3546	0.2936	0.0954	0.1569	0.2438
CODIGEM	0.4126	0.3567	0.2954	0.0948	0.1556	0.2424
CDGAN	**0.4355**	**0.3726**	**0.3084**	**0.1049**	**0.1681**	**0.2635**
Improvement	5.55 %	4.46 %	4.40 %	9.96 %	7.14 %	8.08 %
			Steam			
IRGAN	0.0734	0.0565	0.0373	0.2392	0.3391	0.4165
CFGAN	0.0986	0.0602	0.0370	0.3223	0.3608	0.4038
VCGAN	0.1060	0.0683	0.0421	0.3338	0.3949	0.4482
CODIGEM	0.1128	0.0691	0.0431	0.3454	0.3868	0.4485
CDGAN	**0.1244**	**0.0795**	**0.0504**	**0.3609**	**0.4217**	**0.4906**
Improvement	10.28 %	15.05 %	16.94 %	4.49 %	6.79 %	9.39 %

Table 2. Comparison results of different models on three real datasets. CDGAN is our proposed model, and other models are the comparative baseline models. The best results are shown in bold, the best baseline results are underlined, and the last row is the improvement of CDGAN relative to the best baseline model results.

4.5 Ablation Study

In this section, we will conduct ablation experiments on the feature encoder, the diffusion generator, and the self-attention discriminator to verify the effectiveness of each component of our proposed CDGAN model.

4.5.1 The Effectiveness of Feature Encoder

In Section 3.1, we have provided a detailed introduction to the feature encoder. We use RFE and SIFE to extract features from the user-item rating matrix and user-side information, respectively, and concatenate the extracted latent vectors with the original rating matrix for data augmentation. Table 3 demonstrates the effectiveness of the feature encoder, where CDGAN (without FE) indicates that the feature encoder is not used for feature extraction in the model, and the original rating matrix is directly used as the model's input. The experimental results from Table 3 indicate that the feature encoder component by fully utilizing the side information can enhance the model's performance. We believe that the reason for these results is that the feature encoder has the ability to effectively map high-dimensional matrix into low-dimensional vector, while the generated low-dimensional latent vector skillfully preserves the nonlinear feature of the original matrix. By concatenating L_{UI} and L_{UF} with the original rating matrix, the sparsity of the data can be significantly reduced, thereby improving the effectiveness of the recommendation systems.

Datasets	Metrics	CDGAN (without FE)	CDGAN
MovieLens 100K	Precision@5	0.5756	**0.5917**
	Recall@5	0.1145	**0.1176**
MovieLens 1M	Precision@5	0.4305	**0.4355**
	Recall@5	0.1035	**0.1049**
Steam	Precision@5	0.1222	**0.1244**
	Recall@5	0.3582	**0.3609**

Table 3. Performance comparisons of CDGAN model with and without FE on three datasets. Bold indicates the best results for the listed models.

4.5.2 The Effectiveness of Diffusion Generator

In Section 3.2, we have provided a detailed introduction to the diffusion generator. We employ the DM as the generator in CDGAN to produce a predictive rating matrix that aligns more closely with user preferences. In the forward diffusion process, Gaussian noise is gradually added to the augmented rating matrix, and in the reverse denoising process, the noise is gradually removed to restore the original rating matrix. To validate the effectiveness of the DM in CDGAN, we used other generative models with denoising effects as the generator in CDGAN, such as Denoising Autoencoder (DAE) and Variational Autoencoder (VAE). Specifically, CDGAN_DAE represents the use of a DAE as the generator in CDGAN, and CDGAN_VAE represents the use of a VAE as the generator. Table 4 demonstrates the effectiveness of the diffusion generator, and its results fully reflect the denoising capabilities of the diffusion generator. We believe that the primary reason for this result is that the diffusion generator can successfully capture complex nonlinear patterns in the rating matrix through the forward diffusion and reverse denoising processes. It not only helps to generate strong collaborative signals and reveal the deep connections between users and items but also forms robust latent representations to accurately reflect the characteristics of users and items. Meanwhile, by removing the noise from the input rating matrix, the diffusion generator significantly improves the stability of the model, thus ensuring the quality and reliability of the recommendation results.

Datasets	Metrics	CDGAN_DAE	CDGAN_VAE	CDGAN
MovieLens 100K	Precision@5	0.5098	0.5725	**0.5917**
	Recall@5	0.1035	0.1141	**0.1176**
MovieLens 1M	Precision@5	0.2535	0.4116	**0.4355**
	Recall@5	0.0499	0.0964	**0.1049**
Steam	Precision@5	0.0758	0.1074	**0.1244**
	Recall@5	0.2744	0.3356	**0.3609**

Table 4. Performance comparisons of CDGAN model with DAE, VAE and DM on three datasets. Bold indicates the best results for the listed models.

4.5.3 The Effectiveness of Self-Attention Discriminator

In Section 3.3, we have provided a detailed introduction to the self-attention discriminator. Different from the discriminator of traditional GAN, we have introduced a self-attention mechanism to expand the discriminator into a neural network that can extract explicit user preferences, thus improving the discriminating ability of the discriminator, and then promoting the learning of the generator through the feedback mechanism to improve the generating ability. Table 5 shows the effectiveness of the self-attention discriminator, where CDGAN (without self-attention) refers to the version of its discriminator that does not employ the self-attention mechanism. The experimental results from Table 5 indicate that incorporating a self-attention mechanism into the discriminator of CDGAN enhances the model's performance. We attribute the observed results primarily to the self-attention mechanism's capacity to deeply learn and parse local features within the input data. This characteristic enables the discriminator to more effectively capture and extract features of explicit user preferences, thereby enhancing its ability to distinguish between real data and generated data, and improving the effectiveness of the model's recommendation.

Datasets	Metrics	CDGAN (Without Self-Attention)	CDGAN
MovieLens 100K	Precision@5	0.5882	**0.5917**
	Recall@5	0.1173	**0.1176**
MovieLens 1M	Precision@5	0.4325	**0.4355**
	Recall@5	0.1043	**0.1049**
Steam	Precision@5	0.1242	**0.1244**
	Recall@5	0.3590	**0.3609**

Table 5. Performance comparisons of CDGAN model with and without self-attention mechanism on three datasets. Bold indicates the best results for the listed models.

5 CONCLUSIONS AND FUTURE WORK

This paper proposes a novel recommendation framework named CDGAN and designs the corresponding CDGAN recommendation algorithm. The framework enhances the original rating matrix by extracting the latent feature vectors of ratings and side information through feature encoders to alleviate the data sparsity problem. The noise in data is removed through forward diffusion and reverse denoising processes of the diffusion generator to alleviate the data noise problem. By using the self-attention discriminator to obtain user preferences, the discriminative ability of the discriminator can be improved to help the generator generate more realistic recommendation data. Extensive experiments on three real datasets have verified the effectiveness of the CDGAN proposed in this paper. Compared with the classic GAN-based CF models and DM-based CODIGEM model, our proposed CDGAN model achieves superior performance.

Although CDGAN can effectively alleviate the problems of data sparsity and noise, there are still some limitations. Since we integrated DM as a generator, the sampling speed of the generator became slow. This is mainly because the reverse process of DM relies on a Gaussian distribution approximation for denoising, while the Gaussian assumption is only applicable when the denoising steps are very small. In actual recommendation scenarios, the interactive data of users often has complex distribution characteristics, and it is difficult to accurately recover the data simply by relying on Gaussian distribution denoising. In order to achieve high-quality data recovery, a large number of reverse steps need to be executed. This leads to a long sampling time, which affects the efficiency and practicability of the model in practical applications. Moreover, since the data in recommendation systems often has high dimensionality and complexity, such as user behavior data, item feature data, etc., DM needs to operate and calculate directly in these high-dimensional data spaces, and each iteration needs to process a large amount of data, which also increases the calculation amount and time cost. In future work, we will explore denoising distribution assumptions that are more in line with the actual data distribution or perform dimensionality reduction on the high-dimensional data of the recommendation systems, so as to reduce the dependence on the Gaussian distribution, further reduce the number of denoising steps in the reverse process, and improve the sampling speed.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China under Grant No. 62141203, the Science and Technology Program of Jiangxi Provincial Department of Education under Grant No. GJJ2501519, the Open Project of Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education under Grant No. ESSCKF2024-13, the Jiangxi Provincial Key Laboratory of Electronic Data Control and Forensics under Grant No. 20242BCC32027-8, the Guiding Science and Technology Plan Project of Ji'an

City under Grant No. 20244-029651, the Scientific Research Foundation of Jinggangshan University under Grant No. JAI4S2501.

REFERENCES

[1] Javed, U.—Shaukat, K.—Hameed, I. A.—Iqbal, F.—Alam, T. M.—Luo, S.: A Review of Content-Based and Context-Based Recommendation Systems. International Journal of Emerging Technologies in Learning (iJET), Vol. 16, 2021, No. 3, pp. 274–306, doi: 10.3991/ijet.v16i03.18851.

[2] Xu, W.—Xie, Q.—Yang, S.—Cao, J.—Pang, S.: Enhancing Content-Based Recommendation via Large Language Model. Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), 2024, pp. 4153–4157, doi: 10.1145/3627673.3679913.

[3] Afoudi, Y.—Lazaar, M.—Al Achhab, M.: Collaborative Filtering Recommender System. In: Ezziyyani, M. (Ed.): Advanced Intelligent Systems for Sustainable Development (AI2SD 2018). Springer, Cham, Advanced Intelligent Systems for Computing Sciences, Vol. 915, 2019, pp. 332–345, doi: 10.1007/978-3-030-11928-7_30.

[4] Wang, J.—Yu, L.—Zhang, W.—Gong, Y.—Xu, Y.—Wang, B.—Zhang, P.—Zhang, D.: IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17), 2017, pp. 515–524, doi: 10.1145/3077136.3080786.

[5] Chae, D. K.—Kang, J. S.—Kim, S. W.—Lee, J. T.: CFGAN: A Generic Collaborative Filtering Framework Based on Generative Adversarial Networks. Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18), 2018, pp. 137–146, doi: 10.1145/3269206.3271743.

[6] Wang, H.—Wang, J.—Wang, J.—Zhao, M.—Zhang, W.—Zhang, F.—Li, W.—Xie, X.—Guo, M.: Learning Graph Representation with Generative Adversarial Nets. IEEE Transactions on Knowledge and Data Engineering, Vol. 33, 2021, No. 8, pp. 3090–3103, doi: 10.1109/TKDE.2019.2961882.

[7] Zhou, C. H.—Chen, Y. L.: VCGAN: Variational Collaborative Generative Adversarial Network for Recommendation Systems. ICC 2023 – IEEE International Conference on Communications, 2023, pp. 6324–6330, doi: 10.1109/ICC45041.2023.10278637.

[8] Walker, J.—Zhong, T.—Zhang, F.—Gao, Q.—Zhou, F.: Recommendation via Collaborative Diffusion Generative Model. In: Memmi, G., Yang, B., Kong, L., Zhang, T., Qiu, M. (Eds.): Knowledge Science, Engineering and Management (KSEM 2022). Springer, Cham, Lecture Notes in Computer Science, Vol. 13370, 2022, pp. 593–605, doi: 10.1007/978-3-031-10989-8_47.

[9] Zhu, Y.—Chen, Z.: Variational Bandwidth Auto-Encoder for Hybrid Recommender Systems. IEEE Transactions on Knowledge and Data Engineering, Vol. 35, 2022, No. 5, pp. 5371–5385, doi: 10.1109/TKDE.2022.3155408.

[10] Varasteh, M.—Nejad, M. S.—Moradi, H.—Sadeghi, M. A.—Kalhor, A.: An Improved Hybrid Recommender System: Integrating Document Context-Based

and Behavior-Based Methods. 2023 31st International Conference on Electrical Engineering (ICEE), 2023, pp. 881–887, doi: 10.1109/ICEE59167.2023.10334913.

[11] Luan, W.—Liu, G.—Jiang, C.—Qi, L.: Partition-Based Collaborative Tensor Factorization for POI Recommendation. IEEE/CAA Journal of Automatica Sinica, Vol. 4, 2017, No. 3, pp. 437–446, doi: 10.1109/JAS.2017.7510538.

[12] Luan, W.—Liu, G.—Jiang, C.—Zhou, M.: MPTR: A Maximal-Marginal-Relevance-Based Personalized Trip Recommendation Method. IEEE Transactions on Intelligent Transportation Systems, Vol. 19, 2018, No. 11, pp. 3461–3474, doi: 10.1109/TITS.2017.2781138.

[13] Li, J.—Liu, G.—Yan, C.—Jiang, C.: LORI: A Learning-to-Rank-Based Integration Method of Location Recommendation. IEEE Transactions on Computational Social Systems, Vol. 6, 2019, No. 3, pp. 430–440, doi: 10.1109/TCSS.2019.2907563.

[14] Kompan, M.—Bielikova, M.: Group Recommendations: Survey and Perspectives. Computing and Informatics, Vol. 33, 2014, No. 2, pp. 446–476, `https://www.cai.sk/ojs/index.php/cai/article/view/1077`.

[15] Goodfellow, I. J.—Pouget-Abadie, J.—Mirza, M.—Xu, B.—Warde-Farley, D.—Ozair, S.—Courville, A.—Bengio, Y.: Generative Adversarial Nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. Q. (Eds.): Advances in Neural Information Processing Systems 27 (NIPS 2014). Curran Associates, Inc., 2014, pp. 2672–2680, `https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf`.

[16] Covington, P.—Adams, J.—Sargin, E.: Deep Neural Networks for YouTube Recommendations. Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16), 2016, pp. 191–198, doi: 10.1145/2959100.2959190.

[17] Chen, J.—Zhang, H.—He, X.—Nie, L.—Liu, W.—Chua, T. S.: Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17), 2017, pp. 335–344, doi: 10.1145/3077136.3080797.

[18] Sun, W.—Yu, S.—Dong, B.: VDGAN: A Collaborative Filtering Framework Based on Variational Denoising with GANs. 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8, doi: 10.1109/IJCNN52387.2021.9533585.

[19] He, X.—Liao, L.—Zhang, H.—Nie, L.—Hu, X.—Chua, T. S.: Neural Collaborative Filtering. Proceedings of the 26th International Conference on World Wide Web (WWW '17), 2017, pp. 173–182, doi: 10.1145/3038912.3052569.

[20] Chen, Y.—de Rijke, M.: A Collective Variational Autoencoder for Top-N Recommendation with Side Information. Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems (DLRS 2018), 2018, pp. 3–9, doi: 10.1145/3270323.3270326.

[21] Chen, Y.—Zhao, X.—de Rijke, M.: Top-N Recommendation with High-Dimensional Side Information via Locality Preserving Projection. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17), 2017, pp. 985–988, doi: 10.1145/3077136.3080697.

[22] Ho, J.—Jain, A.—Abbeel, P.: Denoising Diffusion Probabilistic Models. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., Lin, H. (Eds.): Advances in Neural Information Processing Systems 33 (NeurIPS 2020). Curran Associates, Inc., 2020, pp. 6840–6851, `https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf`.

[23] Ferreira, D.—Silva, S.—Abelha, A.—Machado, J.: Recommendation System Using Autoencoders. Applied Sciences, Vol. 10, 2020, No. 16, Art. No. 5510, doi: 10.3390/app10165510.

[24] Ren, R.—Liu, Z.—Li, Y.—Zhao, W. X.—Wang, H.—Ding, B.—Wen, J. R.: Sequential Recommendation with Self-Attentive Multi-Adversarial Network. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), 2020, pp. 89–98, doi: 10.1145/3397271.3401111.

[25] Wasid, M.—Ali, R.: Fuzzy Side Information Clustering-Based Framework for Effective Recommendations. Computing and Informatics, Vol. 38, 2019, No. 3, pp. 597–620, doi: 10.31577/cai_2019_3_597.

[26] Sun, Z.—Guo, Q.—Yang, J.—Fang, H.—Guo, G.—Zhang, J.—Burke, R.: Research Commentary on Recommendations with Side Information: A Survey and Research Directions. Electronic Commerce Research and Applications, Vol. 37, 2019, Art. No. 100879, doi: 10.1016/j.elerap.2019.100879.

[27] Wu, Y.—Macdonald, C.—Ounis, I.: A Hybrid Conditional Variational Autoencoder Model for Personalised Top-n Recommendation. Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval (ICTIR '20), 2020, pp. 89–96, doi: 10.1145/3409256.3409835.

[28] Wei, Y.—Wang, X.—Nie, L.—He, X.—Chua, T. S.: Graph-Refined Convolutional Network for Multimedia Recommendation with Implicit Feedback. Proceedings of the 28th ACM International Conference on Multimedia (MM '20), 2020, pp. 3541–3549, doi: 10.1145/3394171.3413556.

[29] Wen, J.—Chen, B. Y.—Wang, C. D.—Tian, Z.: PRGAN: Personalized Recommendation with Conditional Generative Adversarial Networks. 2021 IEEE International Conference on Data Mining (ICDM), 2021, pp. 729–738, doi: 10.1109/ICDM51629.2021.00084.

[30] Kingma, D. P.—Welling, M.: Auto-Encoding Variational Bayes. CoRR, 2013, doi: 10.48550/arXiv.1312.6114.

[31] Wang, W.—Xu, Y.—Feng, F.—Lin, X.—He, X.—Chua, T. S.: Diffusion Recommender Model. Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), 2023, pp. 832–841, doi: 10.1145/3539618.3591663.

[32] Li, Z.—Sun, A.—Li, C.: DiffuRec: A Diffusion Model for Sequential Recommendation. ACM Transactions on Information Systems, Vol. 42, 2023, No. 3, Art. No. 66, doi: 10.1145/3631116.

[33] Wu, Z.—Wang, X.—Chen, H.—Li, K.—Han, Y.—Sun, L.—Zhu, W.: Diff4Rec: Sequential Recommendation with Curriculum-Scheduled Diffusion Augmentation. Proceedings of the 31st ACM International Conference on Multimedia (MM '23), 2023, pp. 9329–9335, doi: 10.1145/3581783.3612709.

[34] Wang, Y.—Liu, Z.—Wang, Y.—Zhao, X.—Chen, B.—Guo, H.—Tang, R.: Diff-MSR: A Diffusion Model Enhanced Paradigm for Cold-Start Multi-Scenario Recommendation. Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24), 2024, pp. 779–787, doi: 10.1145/3616855.3635807.

[35] Liu, Q.—Yan, F.—Zhao, X.—Du, Z.—Guo, H.—Tang, R.—Tian, F.: Diffusion Augmentation for Sequential Recommendation. Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), 2023, pp. 1576–1586, doi: 10.1145/3583780.3615134.

[36] Dhariwal, P.—Nichol, A.: Diffusion Models Beat GANs on Image Synthesis. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. S., Wortman Vaughan, J. (Eds.): Advances in Neural Information Processing Systems 34 (NeurIPS 2021). Curran Associates, Inc., 2021, pp. 8780–8794, `https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf`.

[37] Rombach, R.—Blattmann, A.—Lorenz, D.—Esser, P.—Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 10674–10685, doi: 10.1109/CVPR52688.2022.01042.

[38] Yu, W.—Li, Z.—Liu, Q.—Chen, Y.—Zhang, S.—Lin, J.: DifBFSR: Blind Face Super-Resolution via Conditional Diffusion Contraction. Computing and Informatics, Vol. 43, 2024, No. 2, pp. 369–392, doi: 10.31577/cai_2024_2_369.

[39] Luo, C.: Understanding Diffusion Models: A Unified Perspective. CoRR, 2022, doi: 10.48550/arXiv.2208.11970.

[40] Vaswani, A.—Shazeer, N.—Parmar, N.—Uszkoreit, J.—Jones, L.—Gomez, A. N.—Kaiser, Ł.—Polosukhin, I.: Attention Is All You Need. In: Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 30 (NIPS 2017). Curran Associates, Inc., 2017, pp. 5998–6008, `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

[41] Kingma, D. P.—Ba, J.: Adam: A Method for Stochastic Optimization. CoRR, 2014, doi: 10.48550/arXiv.1412.6980.

Jinzhong Li received his Ph.D. degree from the Tongji University, China. Currently, he serves as Professor at the School of Electronic and Information Engineering, Jinggangshan University, China. He has authored or co-authored over 100 papers in international conferences and journals. His research interests include recommender systems, information retrieval and machine learning, especially generative adversarial networks and diffusion models.

Junqi Liu is a second-year graduate student at the School of Software Engineering, Jiangxi University of Science and Technology. His research interests include collaborative filtering and sequential recommendation, especially recommendations based on deep generation models.

Zhiqiang Huang is a second-year graduate student at the School of Software Engineering, Jiangxi University of Science and Technology. His research interests include collaborative filtering and diverse recommendation, especially generative recommendations based on generative adversarial networks and diffusion models.