Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

ACDGRec: Adversarial Contrastive Diffusion Graph Neural Network for Robust Recommendation

Yuan Yu
Qufu Normal University
Guangshun Li
guangshunli@qfnu.edu.cn
Qufu Normal University
Junhua Wu
Qufu Normal University
Ruirui Zhang
Qufu Normal University

Research Article

Keywords: Adversarial Training, Contrastive Learning, Diffusion Model, Recommendation System

Posted Date: July 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9812877/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

ACDGRec: Adversarial Contrastive Diffusion Graph Neural Network for Robust Recommendation

Yuan Yu[1], Guangshun Li[1*†], Junhua Wu[1†], Ruirui Zhang[1†]

[1]College of Computer Science, Qufu Normal University, Yan Tai Street, Rizhao, 276826, China.

*Corresponding author(s). E-mail(s): guangshunli@qfnu.edu.cn;
Contributing authors: yuyuan12345678@163.com; shdwjh@163.com;
15224479626@163.com;
†These authors contributed equally to this work.

Abstract

To alleviate the issue of information overload, recommendation systems increasingly leverage deep learning to predict user preferences. Recently, graph-based diffusion generative models have emerged as promising decoding techniques and have been successfully adopted in this domain. Despite their strong generative capacity in modeling user?item interactions, diffusion-based recommendation models still exhibit several limitations. Specifically, they often neglect the latent correlations among user contexts, which hinders their generalization ability, reduces their effectiveness in sparse-data scenarios, and exacerbates the cold-start problem, all while leaving room for improvement in robustness. To address these challenges, we propose an enhanced model that integrates adversarial training and contrastive learning modules to collaboratively boost performance. The contrastive learning module constructs positive and negative sample pairs to improve the model?s ability to discern user interests under low-interaction conditions, thereby effectively mitigating the data sparsity issue. Simultaneously, the adversarial training module introduces perturbations during training to increase robustness to input variations, enhancing model stability when faced with cold-start or anomalous data. Extensive experiments conducted on three real-world datasets demonstrate that our proposed method significantly outperforms several existing recommendation approaches and yields notable improvements over conventional diffusion-based recommendation models.

Keywords: Adversarial Training, Contrastive Learning, Diffusion Model, Recommendation System

1 Introduction

With the rapid proliferation of smart devices and social platforms, user behavior data on the internet has grown increasingly diverse and voluminous. Correspondingly, the data types processed by recommendation systems have expanded beyond traditional explicit ratings to encompass user behavior logs, clickstreams, implicit preferences, and interactions on social networks [1]. To intuitively model these interactions, the core structure of a typical recommendation system is often illustrated as a bipartite graph. As shown in Fig.1, which presents an example consisting of five users and four beverage items, the graph structure effectively captures the latent relationships between users and items. In response to these trends, recommendation systems are transitioning from explicit feedback-based modeling approaches to strategies driven by implicit behaviors, thereby enhancing their dynamic perception and adaptability in personalized services [2, 3].

As a pivotal technology for mitigating information overload, recommendation systems have become indispensable in modern internet and digital platforms. By precisely aligning users with content or products tailored to their preferences, they significantly improve interaction efficiency and user experience. In contexts characterized by information redundancy and decision fatigue, recommendation systems enable users to swiftly focus on relevant content, thereby enhancing the intelligence and responsiveness of platform services.

To effectively cope with rapidly evolving user preferences and increasingly complex behavioral patterns, research in recommendation systems is progressively integrating cognitive modeling, behavioral analysis, and multi-source information fusion. These advancements collectively aim to strengthen the generalization capability and robustness of recommendation algorithms, driving their evolution toward greater intelligence and adaptability.

Diffusion models [4, 5] are a class of deep generative models designed for data generation and modeling. Their core idea originates from the diffusion process in non-equilibrium thermodynamics and typically involves two stages to accomplish the modeling task. The first stage is the forward diffusion process, where Gaussian noise is progressively added to the original data, gradually degrading it into pure noise. The second stage is the reverse denoising process, in which a neural network learns the reconstruction trajectory to recover the original data step by step from the noise. This generative mechanism not only provides high stability and training convergence but also enables the generation of high-quality, structurally complete samples. It has been widely applied in tasks such as image generation, editing, super-resolution, and semantic modeling.

In recommendation systems, diffusion models have also demonstrated great potential. Their core idea lies in simulating the gradual diffusion process of information between users and items to capture deep-level latent preference relations. In graph

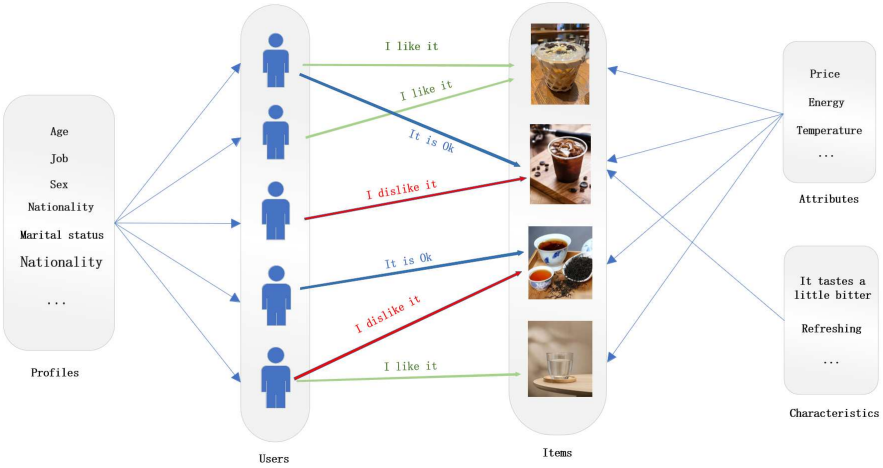


Fig. 1: Illustrative example of a user-item bipartite graph with auxiliary information.

structure modeling and representation learning tasks, diffusion mechanisms have been proven to possess strong generalization ability and adaptability to sparse interactions. They can effectively complete missing information in user-item interaction graphs and enhance the robustness and expressive power of recommendation systems under cold-start and sparse data scenarios. In recent years, this mechanism has been widely applied to graph neural network-based recommendation models, further promoting research and real-world deployment of diffusion models in the field of personalized recommendation.

In 2017, deep generative modeling experienced a surge in development, driven by the success of variational autoencoders (VAEs) [6] and generative adversarial networks (GANs) [7], which spurred extensive research into deep generative architectures. During this time, diffusion models?initially viewed as a variant of generative models?began to emerge in tasks such as image generation. In 2019, diffusion models were formally introduced through the work of Jonathan Ho [8], who defined a generative framework based on a forward diffusion process and a reverse denoising process. In 2020, He X and colleagues proposed the LightGCN model, simplifying the graph convolution network (GCN) for recommendation tasks. This work established diffusion models as a distinct and independent class of generative methods. Since their inception, diffusion models have progressively extended into the domain of recommendation systems. In this setting, sequential user behaviors, item relationships, and other contextual signals can be effectively captured via diffusion processes, thereby enhancing the quality of personalized recommendations.

Early recommendation systems primarily relied on collaborative filtering or matrix factorization methods [9], which typically depend on explicit user feedback for making recommendations. With the advancement of deep learning, diffusion models, as a class of generative models, began to be introduced into recommendation systems?particularly for modeling complex interactions between users and items. In 2022,

diffusion-based recommendation systems emerged as a research hotspot. By simulating the information diffusion process among items, diffusion models assist in capturing latent user preference signals in recommendation tasks. Graph neural networks (GNNs) [10], a deep learning approach capable of handling non-Euclidean data, have achieved remarkable success in recent years and attracted significant research interest.

In recommendation systems, GNNs improve recommendation quality by learning node embeddings through the aggregation of information from neighboring nodes. Building on this capability, GNNs have been incorporated into diffusion models by designing a graph-based decoding network (GDN) [11] specifically for the reverse diffusion process. This network includes architectural components such as skip connections and normalization layers, which contribute to enhanced recommendation performance. Within this generative framework, item embeddings are generated more effectively, and the overall message propagation efficiency is significantly improved. Nevertheless, existing models still face several limitations.

1) Limited capability in capturing long-range dependencies. The graph-based decoding network relies on graph neural networks for information propagation. However, when multiple GNN layers are stacked, the model tends to suffer from oversmoothing, where node representations become excessively similar. This phenomenon undermines the ability to capture semantic dependencies between distant nodes. Deep GNN architectures are susceptible to the vanishing gradient problem [12], where gradients diminish as the network depth increases, thereby impairing the model?s efficiency in learning long-range structural information.

2) Limited ability to handle sparse connections. In practical graph data, sparsity is prevalent, which may hinder effective information propagation and representation learning within the GDN. The model?s ability to capture meaningful user-item interactions is compromised under sparse connectivity patterns.

3) Vulnerability to adversarial manipulation. In the presence of adversarial attacks, recommendation systems can be intentionally manipulated, leading to inaccurate or even harmful recommendation results. Such vulnerabilities may be exploited to promote undesired content, distort user preferences, or degrade the overall trustworthiness of the system.

To address the aforementioned limitations, we incorporate contrastive learning and adversarial training modules into the model, which collaboratively mitigate data sparsity and enhance robustness. Specifically, the contrastive learning module improves data representation by constructing positive and negative sample pairs to distinguish between similar and dissimilar semantics, enabling the capture of latent relational structures even in sparse scenarios to enhance generalization. Simultaneously, the adversarial training module refines the generative process through an adversarial interplay between the generator and the discriminator [13]; by introducing perturbations and synthesizing complex negative instances, it bolsters the model?s robustness against noise and forces the generator to better approximate real user distributions, effectively balancing accuracy with diversity.

In summary, our main contributions are as follows.

1. We propose ACDGRec, a novel recommendation framework that organically integrates Gaussian diffusion processes, graph neural networks, contrastive learning,

and adversarial training. This synergistic design effectively addresses core challenges in recommendation systems, specifically targeting data sparsity and insufficient model robustness.

2. To enhance model expressiveness in sparse interaction scenarios, we design a contrastive learning module based on the InfoNCE loss. By constructing positive and negative sample pairs, this module reinforces the discriminability of user and item representations, effectively alleviating performance degradation caused by limited data.
3. To improve stability in non-ideal environments, we incorporate an adversarial training mechanism that introduces input-space perturbations. This mechanism significantly enhances the model?s adaptability to abnormal behaviors and cold-start scenarios, ensuring consistent performance even under noise or potential attacks.

The rest of this article is organized as follows. Section 2 reviews related work. Section 3 details the proposed methodology. Section 4 presents the experimental results. Section 5 concludes the paper.

2 Related work

2.1 Graph-based diffusion recommendation models

Since the beginning of the 21st century, neural networks have achieved significant progress, however, they initially lacked effective methods for handling graph-structured data. Traditional neural networks were still limited in their ability to model such data. To address this limitation, researchers proposed graph neural networks, aiming to integrate neural architectures with graph-structured inputs and enable automatic modeling of relationships between graph nodes, thereby learning latent patterns in graph data.

In 2005, Scarselli [14] introduced the first GNN model, which leveraged recurrent neural networks (RNNs) to process each node in the graph while accounting for inter-node information propagation. GNNs are capable of handling arbitrary graph structures by combining node features with those of their neighbors, thus enabling effective processing of graph data. Due to limited computational resources and theoretical foundations at the time, this approach did not immediately gain widespread adoption.

In 2017, Thomas Kipf and Max Welling introduced the GCN [15], which laid the groundwork for applying GNNs to recommendation systems. Researchers began leveraging GCNs to model user-item interaction graphs, learning latent representations by propagating and aggregating information across graph structures. Specifically, GCNs update user and item embeddings using neighboring nodes? features, effectively capturing latent collaborative filtering signals.

PinSage [16], proposed by Facebook, is a scalable recommendation system that integrates graph convolution with deep learning techniques to address recommendation challenges in large-scale social networks. Experimental results showed that structural information from graphs can significantly improve recommendation accuracy. This

approach is particularly effective in handling sparse user-item matrices and is well-suited for tasks involving social relationships or inter-item dependencies. Then, He X and colleagues proposed the LightGCN model [17], simplifying the graph convolution network for recommendation tasks.

With the ongoing development of GNN architectures, numerous variants have emerged to meet specific needs in recommendation scenarios. In 2018, graph attention networks (GATs) [18] were proposed. GAT employs a self-attention mechanism to learn adaptive weights for different neighbors, enabling dynamic aggregation of neighborhood information. In recommendation tasks, this mechanism allows the model to assign varying importance to different neighbors, overcoming the uniform weighting limitation in traditional GCNs and significantly enhancing the flexibility of graph-based representation learning.

GraphSAGE [19] introduced a novel sampling-based graph convolution framework that selects a fixed-size set of neighbors to enable efficient learning on large-scale graphs. In recommendation systems, GraphSAGE is particularly advantageous for massive user-item interaction graphs, as it allows for scalable training and inference without requiring access to all neighboring nodes. KGCN [20] generates item embeddings using GCNs from collaborative filtering knowledge graphs, but its user embeddings do not rely on the knowledge graph.

In contrast, KGAT [21] integrates GAT with the TransR model to incorporate the topological structure of the knowledge graph into the embeddings, thereby enhancing recommendation performance. CGAT [22] employs a strategy that combines random walks with RNNs to capture higher-order neighbor information for items in the knowledge graph, enabling the model to identify more complex item relationships. The subsequent model, CODIGEM [23], is a graph-based neural network approach aimed at improving recommendation system performance using deep learning techniques. This model addresses key challenges in recommendation systems, including sparsity, cold-start issues, and the dynamic modeling of user behavior. Following that, DiffuRec [24] reconstructs item embeddings generated by Transformer models to enhance recommendation quality. DreamRec [25] introduces a guiding signal that controls the generation of user interaction sequence embeddings.

With the rapid growth of deep learning, diffusion models have progressively expanded from fields like image generation and language modeling to recommendation systems. These models are particularly effective in capturing complex, sparse interactions between users and items. User behavior in recommendation systems is often characterized by high uncertainty and dynamic patterns, and diffusion models, by simulating gradual perturbation and recovery of information, offer superior generative capabilities. They are thus well-suited for modeling the evolution of latent user preferences and generating personalized recommendations.

Recently, G-Diff [26], proposed by Chen et al., is an innovative recommendation system that combines diffusion processes with GNNs. The G-Diff model aims to model complex user-item interactions by simulating the gradual diffusion of noise. Through the integration of diffusion mechanisms, G-Diff progressively generates high-quality personalized recommendation lists, exhibiting strong generative power and robustness. It can effectively handle diverse and sparse recommendation scenarios. Furthermore,

G-Diff leverages graph structural information to capture both global and local user behavior features, and it enhances recommendation performance via an adversarial training mechanism, producing more realistic and relevant recommendations.

2.2 Adversarial Training for Recommendation Models

In recent years, adversarial training has become a key technique for enhancing model robustness, expanding from traditional domains such as image recognition to recommendation systems. Given that deep learning models are often sensitive to input perturbations, researchers have explored min-max optimization strategies to increase the reliability and generalization ability of models in real-world environments. In 2013, Christian Szegedy et al. introduced the concept of adversarial examples [27], highlighting the vulnerability of deep neural networks to input disturbances, which expose structural weaknesses in the models. Building upon this, Aleksander Madry et al. proposed the classic projected gradient descent adversarial training framework [28], which significantly improved model robustness against various types of attacks, and is now considered the standard approach for robust optimization.

In 2018, He X and colleagues introduced the Adversarial Personalized Ranking method [29], which integrated adversarial training into personalized ranking tasks. This method extends the traditional Bayesian Personalized Ranking framework [30] by adding adversarial perturbations to improve the latent representations of users and items, thereby enhancing the model?s robustness and generalization.

With the growing application of GNNs in recommendation systems, social recommendation tasks have garnered increasing research attention. In this context, Yu et al. proposed the socially-aware neural graph collaborative filtering (SNGCF) model [31], which combines graph-based collaborative filtering with social modeling techniques. The model leverages structural information from social relationship graphs to enhance user preference modeling. Unlike traditional methods, SNGCF explicitly incorporates social graphs, propagates user neighborhood information through multi-layer GNNs, and applies adversarial training to improve robustness against noise in social relationships. This research demonstrates that the integration of adversarial training with graph-based modeling offers substantial improvements in both recommendation accuracy and robustness.

2.3 Contrastive Learning for Recommendation Models

Contrastive learning has recently become one of the leading techniques in unsupervised representation learning. Its core idea is to bring positive samples closer together and push negative samples further apart in the latent space, thereby improving the discriminative capacity of feature representations. The earliest contrastive learning framework can be traced to the DrLIM model proposed by R. Hadsell [32], which utilized contrastive loss to effectively learn image embeddings, establishing the theoretical foundations of contrastive learning. Aaron van den Oord systematically introduced the InfoNCE loss function in his work on contrastive predictive coding (CPC), combining mutual information maximization with contrastive loss, thereby providing theoretical support for the broadening of contrastive learning across various domains.

In 2018, Aaron van den Oord and colleagues introduced a general method that combined autoregressive models with contrastive loss [33], providing a novel approach for unsupervised learning of high-dimensional data. This CPC framework predicts future latent variables and constructs positive and negative sample pairs, aiming to maximize their mutual information. By extracting abstract features that are highly discriminative for downstream tasks, the method reduces the dependency on labeled data and shows superior transferability across different modalities, such as speech, images, text, and reinforcement learning. This approach effectively mitigates the challenges faced by supervised learning, particularly in data-scarce situations. Further advancements explored the synergy between graph structures and contrastive learning. In 2021, Jiancan Wu introduced a structure-aware contrastive learning method for graph-based recommendation systems [34], incorporating local structure perturbations and node-level contrastive objectives to improve the robustness of user and item representations against neighborhood structural changes.

At present, contrastive learning has demonstrated significant potential in recommendation systems, particularly in enhancing the model?s ability to capture shifting user preferences and improving performance for cold-start users. However, most existing approaches are still focused on single-view or shallow contrastive objectives, often overlooking the evolution of interaction relationships in dynamic settings.

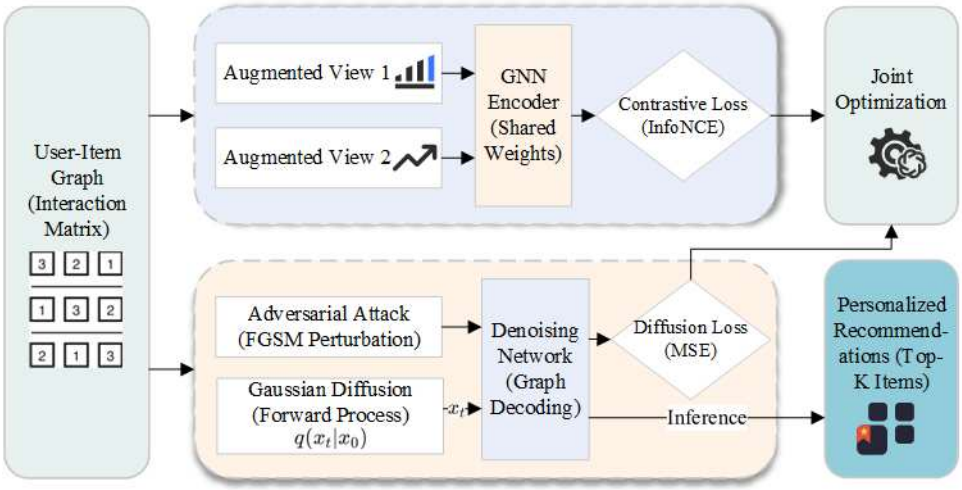


Fig. 2: The overall architecture of the proposed ACDGRec framework.

3 Model Method

3.1 Method Overview

In this section, we introduce the ACDGRec framework, a robust generative recommendation architecture designed to unify structural geometry, semantic consistency,

and adversarial stability. The philosophy of ACDGRec transcends the simple combination of disparate modules; it is built upon a synergistic logic where each component compensates for the inherent limitations of the others in sparse and noisy environments.

Generative foundation with structural priors: At its core, ACDGRec utilizes a Gaussian diffusion process as the generative engine to model the manifold of user-item interactions. To ensure that the generative trajectory respects the underlying relational topology, we integrate a GNN-based denoiser, which treats the denoising task as a structural recovery problem on the interaction graph.

Semantic anchoring via contrastive learning: A major challenge in diffusion-based models is the risk of "generative drift" in sparse data scenarios, where limited supervision fails to constrain the latent representations. To address this, our contrastive learning module acts as a semantic anchor. By maximizing mutual information between augmented views, it enforces discriminative clustering, ensuring the diffusion model remains focused on stable user preference patterns even when interaction signals are faint.

Robustness regularization via adversarial training: To further shield the generative process from anomalous behaviors or malicious perturbations, we incorporate an adversarial training mechanism. Instead of being a mere defensive add-on, this module serves as a robustness regularizer for the diffusion trajectory. By simulating worst-case input variations, it forces the GNN denoiser to learn a smoother and more resilient embedding space, thereby bridging the gap between high accuracy and system reliability.

The entire framework is unified through a joint optimization process, which minimizes the total loss L_{Total} to balance generative fidelity, semantic discriminability, and structural robustness.

3.2 Graph-Based Gaussian Diffusion Framework

The current mainstream diffusion model methods can be categorized into three core frameworks: denoising diffusion probabilistic models [35], noise conditional score networks [36], and general extensions based on stochastic differential equations [37]. Our model builds upon the Gaussian diffusion framework to generate noise and simulate the data diffusion process. As illustrated in Fig. 3, the generative process consists of two stages: the forward process, which gradually adds noise to the original data, and the reverse process, where the model learns to remove the noise and restore the original data [38, 39]. To effectively capture the complex structural relationships between users and items during this process, we integrate GNNs into the reverse denoising stage.

1) Forward Diffusion Process

The forward diffusion process is defined as a Markov chain that progressively adds Gaussian noise to the original data x_0 until it transforms into pure noise x_t. The diffusion process at each step is described by:

$$x_t = \sqrt{\alpha_t}\, x_{t-1} + \sqrt{1-\alpha_t}\, \epsilon_t \tag{1}$$

where x_t denotes the sample at the step t, and α_t is the noise schedule parameter, ϵ_t represents the noise sampled from the standard Gaussian distribution $\mathcal{N}(0, I)$.

This process progressively adds noise, gradually blurring the data until it eventually becomes completely random noise.

2) Reverse Denoising with GNNs

The reverse diffusion process is the core of the model, focusing on restoring the original data from noise. It is described by the following equation:

$$p(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I) \tag{2}$$

where $\mu_\theta(x_t, t)$ is the denoising parameter estimated by the neural network model, representing the predicted mean of the previous state x_{t-1} at step t based on the sample x_t. σ_t^2 is the standard deviation of the diffusion process, typically treated as a learnable parameter.

In our framework, we employ GNNs to parameterize this denoising function. Users and items are represented as nodes, and interactions are represented as edges. By aggregating information from neighboring nodes via graph convolution operations, the GNN updates the representation of each node. This structural encoding enables the model to effectively process sparse user-item interaction data and accurately predict the mean of the previous state x_{t-1}, thereby guiding the generation of high-quality recommendation lists.

A crucial theoretical insight is that adversarial perturbations do not disrupt the original generative distribution; rather, they provide boundary regularization for the denoising network. By minimizing L_{adv} in Eq. 10, the model ensures that the predicted mean μ_θ remains stable even when the input x_t is pushed slightly off the high-density manifold. This synergy allows the diffusion process to bridge the gap between global distribution modeling and local structural robustness, particularly in cold-start scenarios where interaction signals are inherently noisy.

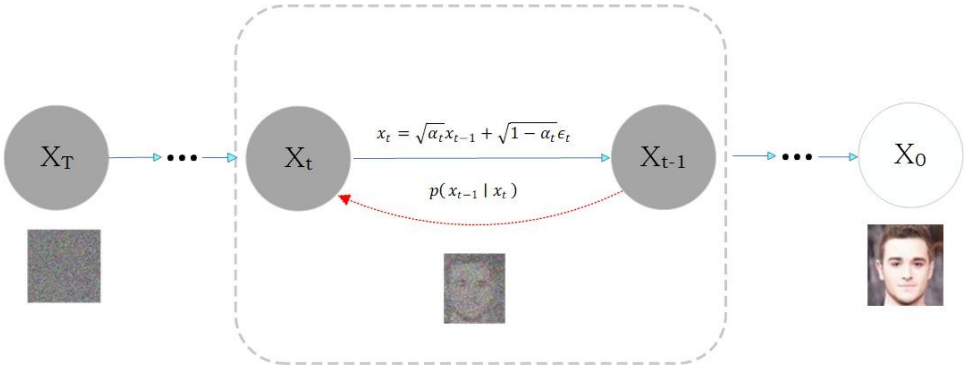


Fig. 3: Schematic illustration of the forward and reverse processes in the Gaussian diffusion model.

3.3 Graph-Aware Contrastive Learning

To address the data sparsity issue and provide structural anchors for the diffusion process, we design a graph-aware contrastive learning module. Unlike generic contrastive learning that applies random noise to embeddings, our module performs augmentations directly on the user-item interaction graph $G = (V, E)$ to capture robust structural patterns.

1) Structural and Feature Augmentations.

We generate two correlated augmented views, $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, by applying stochastic perturbations to the graph structure and node features.

Edge Dropping: We randomly remove a portion of the observed edges with a dropout rate ρ_e. This simulates the scenario of missing interactions, forcing the model to infer user preferences from incomplete structural information. Formally, $E_1 = \{(u, i) \in E \mid m_{u,i} = 1\}$, where $m_{u,i} \sim \text{Bernoulli}(1 - \rho_e)$.

Feature Masking: For the second view, we apply a masking operator to the node initial embeddings, where each dimension is set to zero with probability ρ_f. This prevents the model from over-relying on specific feature dimensions and improves representation generalizability.

The two augmented views are processed by the shared GNN encoder $f_\theta(\cdot)$ to obtain high-order representations:

$$z_u^{(1)} = f_\theta(u, G_1), \quad z_u^{(2)} = f_\theta(u, G_2) \tag{3}$$

where $z_u^{(1)}$ and $z_u^{(2)}$ are the latent embeddings of user u under different structural perturbations.

2) Generation of Embedding Representations.

The two augmented views are fed into the model to obtain their corresponding embedding representations z_i and z_j:

$$z_i = f(x_i) \tag{4}$$

$$z_j = f(\tilde{x}_i) \tag{5}$$

3) Similarity Matrix Computation.

All embedding representations within the batch are aggregated, and their pairwise cosine similarities are computed:

$$\text{sim}(z_i, z_j) = \frac{z_i^\top z_j}{\|z_i\| \|z_j\|} \tag{6}$$

4) InfoNCE Loss Computation.

The InfoNCE loss aims to maximize the similarity between positive sample pairs while minimizing the similarity between negative sample pairs. It is defined as follows:

$$L_{\text{InfoNCE}} = -\log \frac{\exp\big(\text{sim}(z_i, z_j)/\tau\big)}{\sum_{k=1}^{2N} \mathbf{1}_{[k \neq i]} \exp\big(\text{sim}(z_i, z_k)/\tau\big)} \tag{7}$$

where z_i and z_j denote the embedding representations of two augmented views originating from the same sample. τ is the temperature parameter that controls the concentration level of the similarity distribution. N refers to the batch size; since each sample produces two augmented views, there are a total of 2N embeddings. The indicator $k \neq$ i ensures that self-similarity is excluded from the denominator. The numerator captures the similarity between the positive pair, while the denominator aggregates the similarities across all possible pairs, including both positive and negative samples. By minimizing the InfoNCE loss, the model is driven to increase the similarity of positive pairs while decreasing that of negative pairs.

3.4 Adversarial Training

To ensure the stability of the generative process, we incorporate an adversarial training mechanism that acts as a structural smoother for the diffusion trajectory. Unlike standard diffusion models that only account for random Gaussian noise, our module explores the 'worst-case' perturbations within the embedding manifold, forcing the GNN-based denoiser to learn a more resilient reverse process $p_\theta(x_{t-1}|x_t)$.

1) Generation of Adversarial Examples

We employ the fast gradient sign method (FGSM) [40] to construct adversarial examples. FGSM introduces perturbations to the normalized user-item embeddings in the direction of the loss gradient, thereby maximizing the model's loss.While more complex iterative attack methods exist, we specifically adopt the FGSM for its computational efficiency and first-order gradient approximation capabilities. In the context of large-scale recommendation graphs, FGSM provides a rapid yet effective way to probe the model?s vulnerability without significantly increasing the training overhead, making it ideal for regularizing high-dimensional embedding spaces. The adversarial perturbation δ is formulated as:

$$\delta = \epsilon \cdot \text{sign}(\nabla_x L_{\text{diff}}(x, y, \theta)) \tag{8}$$

$$x' = x + \delta \tag{9}$$

where x represents the original embedding input, x' is the generated adversarial example, and ϵ controls the magnitude of the perturbation. $\nabla_x L_{\text{diff}}$ denotes the gradient of the diffusion loss with respect to the input x, indicating the direction that is most detrimental to the model's prediction.

2) Adversarial Loss Component

After generating the perturbed sample x', we compute the adversarial loss L_{adv} by feeding x' into the diffusion model. The objective of this component is to minimize the recommendation error even when the input is corrupted:

$$L_{\text{adv}}(x', y, \theta) = L_{\text{diff}}(x + \delta, y, \theta) \tag{10}$$

By minimizing the InfoNCE loss, the model is driven to establish a discriminative representation manifold where semantically similar users are clustered together despite sparse interaction signals. This provides a stable semantic foundation for the

subsequent generative process, significantly enhancing the model's ability to recover latent preferences in data-scarce scenarios.

Furthermore, the integration of graph-aware contrastive learning serves as a topological constraint for the Gaussian diffusion process. While the diffusion mechanism explores the complex generative space, the contrastive objective ensures that the model preserves intrinsic semantic identities across different structural views. This synergy effectively mitigates the risk of 'generative drift' in sparse datasets like Beauty and Toys, where traditional diffusion models often suffer from performance degradation due to insufficient supervision.

3.5 Optimization and Loss Function

To enable end-to-end optimization of the ACDGRec framework, we formulate a joint objective function that integrates the recommendation task, adversarial defense, and contrastive regularization.

1) Joint Loss Formulation

The final total loss function L_{Total} consists of three components: the standard diffusion loss on clean data, the adversarial loss on perturbed data, and the contrastive learning loss. The unified objective is formulated as follows:

$$\begin{aligned} L_{\text{Total}} = & \underbrace{\lambda_{\text{adv}} L_{\text{diff}}(x, y, \theta) + (1 - \lambda_{\text{adv}}) L_{\text{diff}}(x', y, \theta)}_{\text{Robust Diffusion Loss}} \\ & + \underbrace{\lambda_{\text{cl}} L_{\text{InfoNCE}}}_{\text{Contrastive Regularization}} \end{aligned} \tag{11}$$

where $L_{\text{diff}}(x, y, \theta)$ represents the standard Gaussian diffusion loss computed on the original user-item interaction data x, and $L_{\text{diff}}(x', y, \theta)$ denotes the adversarial loss computed on the perturbed samples x' generated via the adversarial training module. L_{InfoNCE} is the contrastive loss derived from the data augmentation views, aiming to maximize the mutual information between positive pairs. The parameter $\lambda_{\text{adv}} \in [0, 1]$ is a balancing coefficient that controls the trade-off between standard training and adversarial defense, while λ_{cl} is a hyperparameter that governs the weight of the contrastive learning task in the overall optimization objective. This joint loss formulation L_{Total} ensures that the diffusion model does not merely memorize the training interactions but learns a robust manifold that is invariant to both random noise and intentional perturbations.

The balancing coefficients λ_{adv} and λ_{cl} govern the trade-off between standard generative accuracy and the auxiliary regularization constraints. To ensure optimal performance and stability, these hyperparameters are determined via a grid search over the range $\{0.01, 0.1, 1, 10\}$ based on the Recall@20 performance on the validation set. This data-driven selection process ensures that the model effectively integrates contrastive and adversarial signals without overwhelming the primary recommendation task. A detailed sensitivity analysis of these parameters is provided in Section 4.5 to further validate the model's stability. The entire framework is trained end-to-end by minimizing the joint loss L_{Total} using the AdamW [41] optimizer.

3.6 Inference Procedure

To provide a clear understanding of how ACDGRec performs recommendations in a real-world online environment, we outline the comprehensive inference procedure below. Unlike traditional discriminative models, our framework leverages a progressive generative process to refine user preference representations.

Online Inference Process of ACDGRec

Require: User u, Candidate item set $\mathcal{I}_{cand}$, Pre-trained Multi-view Graph G, Denoising steps T.
Ensure: Top-K recommendation list L_u.
Initialize: Obtain user representation $\mathbf{z}_u$ and candidate item embeddings $\mathbf{E}_{\mathcal{I}}$ from the graph encoder.
Set initial state $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ or use the encoded prior.
for $t = T, T-1, \ldots, 1$ **do**
    Compute graph-aware semantic context from multi-view decoding layers.
    Predict the noise or denoised representation: $\mathbf{x}_{t-1} = \text{Denoise}(\mathbf{x}_t, \mathbf{z}_u, G, t)$.
    Apply structural constraints to align $\mathbf{x}_{t-1}$ with the graph manifold.
end for
Scoring: Calculate matching scores $S_{u,i} = \phi(\mathbf{x}_0, \mathbf{e}_i)$ for all $i \in \mathcal{I}_{cand}$, where ϕ is a similarity function (e.g., inner product).
Ranking: Sort candidate items in descending order based on $S_{u,i}$.
Generate: $L_u \leftarrow$ select top-K items from the sorted list.
return L_u

As illustrated in the outlined procedure, the inference of ACDGRec transforms the recommendation task into a progressive generative process. The procedure consists of three core phases. First, we initialize the user?s preference state by combining latent representations from the pre-trained graph encoder with Gaussian noise. Second, the model performs iterative denoising for T steps, where multi-view semantic context is extracted to guide the latent vector toward the structural manifold of the graph. Finally, the reconstructed clean preference representation $\mathbf{x}_0$ is matched against the candidate set using a similarity function ϕ to generate the final Top-K recommendation list.

3.7 Coupling Analysis of ACDGRec

To further elucidate the mechanism-level contributions of ACDGRec, this section analyzes the internal coupling and synergistic effects among its four core components: the Gaussian diffusion process, GNN-based denoising, graph-aware contrastive learning, and adversarial training.

The Gaussian diffusion process serves as the generative engine, while the GNN acts as its architectural backbone. In the reverse denoising stage, the GNN does not merely perform feature extraction; it serves as a structural prior that guides the restoration of original data x_0 from noise. By aggregating neighborhood information, the GNN

ensures that the generated recommendation trajectory respects the underlying relational topology of the user-item interaction graph, thereby improving the structural fidelity of the recommendations.

In sparse data scenarios, diffusion models often suffer from generative drift, where the lack of supervision causes the denoising trajectory to deviate from the true user preference manifold . The integration of graph-aware contrastive learning functions as a semantic anchor. By maximizing the mutual information between different structural views, the InfoNCE loss enforces a discriminative constraint on the latent space. This ensures that even when interaction signals are faint, the diffusion process remains focused on stable, intrinsic semantic identities, effectively bridging the gap between generative exploration and representation accuracy.

The adversarial training module serves as a boundary regularizer for the entire generative framework. While standard Gaussian diffusion accounts for random noise, adversarial perturbations δ probe the most vulnerable regions of the embedding space . By minimizing the adversarial loss L_{adv}, the GNN denoiser is regularized to produce consistent outputs within an ϵ-neighborhood of the clean manifold. This synergy ensures that the reverse denoising process is not only accurate but also smooth and resilient to anomalous behaviors, providing a robust convergence guarantee for the joint optimization objective L_{Total}.

4 Experiments

4.1 Experimental Objectives and Evaluation Metrics

In this section, we present a series of experiments designed to validate the following aspects of our model.

1) The relevance, quality, and accuracy of the recommendation results are assessed using the normalized discounted cumulative gain (NDCG) metric. The recall rate is computed using the following formula:

$$\text{Recall@}K = \frac{|R_u \cap S_u^{(K)}|}{|R_u|} \tag{12}$$

where R_u denotes the set of items that user u is actually interested in. $S_u(K)$ represents the set of top-K items recommended to user u. $|R_u \cap S_u(K)|$ denotes the number of relevant items appearing in the top-K recommendations. $|R_u|$ represents the total number of relevant items for user u.

2) The recall rate is used to evaluate the model?s generalization ability and its effectiveness in covering items that align with the user's actual interests. The NDCG metric comprises two parts. The first part, discounted cumulative gain at rank K (DCG@K), is computed as follows:

$$\text{DCG@K} = \sum_{i=1}^{K} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)} \tag{13}$$

where rel_i denotes the relevance score of the item ranked at position i.

The second component, ideal discounted cumulative gain at rank K (IDCG@K), is computed using the following formula:

$$\text{IDCG@K} = \sum_{i=1}^{z} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)} \tag{14}$$

where Z denotes the set of items that the user is genuinely interested in. Based on this, the final NDCG at rank K is determined by:

$$\text{NDCG@K} = \frac{\text{DCG@K}}{\text{IDCG@K}} \tag{15}$$

3) The mean reciprocal rank (MRR) is employed to assess the relevance of results in search and recommendation systems, specifically evaluating the rank quality of the first correctly recommended item in the result list. The MRR metric is computed using the following formula:

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}_i} \tag{16}$$

where N denotes the total number of queries. rank_i represents the reciprocal rank for the i-th query, which corresponds to the position of the first relevant result in the ranked list. If no relevant result exists for the i-th query, its reciprocal rank is set to 0, or the query is excluded from the MRR computation.

4.2 Experimental Setup

To comprehensively evaluate the performance of ACDGRec, we conducted experiments on three widely used benchmarks and compared our method against several state-of-the-art baselines. In this section, we detail the datasets, baseline models, and implementation parameters used in our study.

4.2.1 Datasets

We evaluate our model on three real-world datasets: MovieLens-1M, Beauty, and Toys. MovieLens-1M is a widely adopted benchmark dataset for recommendation system research, collected from the MovieLens website, comprising approximately one million user ratings of movies. Beauty contains user interaction records related to beauty products from Amazon. Toys is a publicly available dataset from Amazon, which includes user reviews, ratings, and metadata for toys and games.

4.2.2 Baseline Models

To comprehensively evaluate the performance of ACDGRec, we select nine representative baseline models for comparison, covering three distinct categories: matrix factorization, graph neural networks, and generative diffusion models.

1. NGCF [42]: It is a recommendation approach based on GNN. In the user-item interaction graph, it leverages a layer-wise feature propagation mechanism to capture potential higher-order collaborative signals by aggregating node information, thereby enhancing the representation power of both users and items. NGCF significantly improves recommendation accuracy and is well-suited for modeling complex graph structures.
2. LightGCN [17]: This model is a lightweight variant of NGCF. It eliminates the nonlinear activation functions and feature transformation operations, retaining only the linear aggregation of neighboring node information. This simplification reduces computational complexity significantly while maintaining efficient collaborative representation learning, making it particularly well-suited for large-scale recommendation systems.
3. CIGCN [43]: By integrating user interest modeling with a collaborative signal propagation mechanism, CIGCN incorporates interest context information into the user-item graph, thereby enhancing the model's ability to capture dynamic interests and personalized features. CIGCN effectively captures local interest shifts and higher-order adjacency features, thereby improving the model?s personalized recommendation performance.
4. CVGA [44]: This model integrates collaborative filtering with variational graph autoencoders. By combining collaborative filtering principles with a graph-based variational encoding mechanism, it learns the probabilistic representation distributions of users and items in the interaction graph. This fusion enhances the joint modeling of collaborative signals and structural relationships, thereby improving the representational capacity and generative performance of the recommendation system.
5. DiffRec [1]: This model is a generative recommendation approach based on diffusion probability modeling. Its core mechanism simulates the forward and backward diffusion processes of interactions between users and items, recovering latent representations from noisy data to generatively model user interests. DiffRec exhibits high recommendation accuracy and generalization capabilities, particularly excelling in modeling latent preferences.
6. DiffRec-L [1]: This model is an extended diffusion model recommendation algorithm that further enhances the recommendation performance of long-tail items, building upon DiffRec. By designing specific diffusion step sizes and perturbation strategies for less popular items, DiffRec-L improves overall accuracy while significantly boosting its ability to capture low-frequency interaction data, ensuring a balanced recommendation of both mainstream items and long-tail content.
7. G-Diff [26]: By deeply integrating graph structure modeling with a diffusion generation mechanism, this method emphasizes learning the semantic representations of users and items within the graph through the diffusion process. It introduces multi-order diffusion paths and a structure-aware mechanism, effectively enhancing the model?s ability to capture complex interaction relationships, and demonstrates robust stability and expressive power across a variety of recommendation tasks.
8. GCL-DT [2]: This model integrates graph contrastive learning with diffusion-based transfer for cross-domain recommendation. It bridges source and target domains

through a diffusion process to transfer latent structural patterns. GCL-DT excels in handling extreme sparsity and cold-start scenarios via robust knowledge migration.

9. DimeRec [25]: DimeRec is a generative sequential recommendation approach based on diffusion models. It optimizes the noise-scheduling strategy to capture evolving temporal patterns and user interests. DimeRec demonstrates superior stability in modeling long-tail behaviors and enhancing next-item prediction accuracy.

4.2.3 Implementation Details

For GCL-DT, NGCF, and CIGCN, the embedding dimension is set to 64, with a default learning rate of 0.001. We optimize all models using the AdamW optimizer, and the weight decay is searched over the set 10, 1, 0.1, 0.01, 0.001 to prevent overfitting. For CIGCN, the dropout ratio is searched within the set 0.1, 0.3, 0.5, and the number of graph message propagation layers is explored within the set 1, 2, 3.

LightGCN: The embedding size is set to 64, with a default learning rate of 0.001 and a default batch size of 1024. The L_2 regularization coefficient λ is searched over the range 1e-6, 1e-5, ..., 1e-2.

G-Diff: We reproduce the model based on its optimal settings. For the MovieLens-1M dataset, we conduct tests with a single graph decoding layer set to 1. In the Beauty dataset, the number of graph decoding layers is set to 6. For the Toys dataset, the number of graph decoding layers is set to 3 and 4, and the optimal result is selected as the baseline for comparison.

Multi-dae and ,DimeRec: The hidden size is set to [1000, 500]. The default encoder and decoder are MLP. Dropout is used to prevent overfitting, and the dropout ratio is searched within the range 0.1, 0.3, 0.5.

DiffRec and DiffRec-L: The step size embedding is fixed at 10. The default encoder and decoder are MLP. The hidden size for $p_\theta(x_{t-1} \mid x_t)$ is selected from [300], [200,600], and 1000. The diffusion step length T an inference step length T are searched within { 2, 5, 10, 40, 50, 100 }. The noise scale S, noise lower bound $\alpha_{\min}$, and noise upper bound $\alpha_{\max}$ are searched over $\{0, 1e^{-5}, 1e^{-4}, 5e^{-3}, 1e^{-2}, 1e^{-1}\}$, $\{5e^{-4}, 1e^{-3}, 5e^{-3}\}$, and $\{5e^{-3}, 1e^{-2}\}$, respectively.

During training, each model is evaluated every 5 epochs. If Recall@20 is not improved within 20 epochs, training is halted, and the recommendation metric corresponding to the optimal Recall@20 is selected as the final result, thus preventing overfitting.

4.3 Overall Performance Comparison

The comparative results of ACDGRec and the baseline models on the three datasets are reported in Table 1, Table 2 and Table 3, respectively. In these tables, the best results for each metric are highlighted in bold.

As observed from the tables, ACDGRec demonstrates highly competitive performance across all datasets and outperforms the state-of-the-art baselines on the vast majority of evaluation metrics, exhibiting particularly significant advantages in sparse scenarios. Specifically, on the MovieLens-1M dataset, ACDGRec achieves Recall@10 and Recall@20 values of 13.17% and 20.65%, respectively, improving by 2.00% and

Table 1: Scores on MovieLens-1M

Metric	R@10	R@20	N@10	N@20	M@10	M@20
GCL-DT	0.1288	0.2053	0.1095	0.1346	0.1732	0.1856
NGCF	0.0768	0.1309	0.0704	0.0886	0.1262	0.1371
LightGCN	0.0957	0.1637	0.0813	0.1044	0.1334	0.1457
CIGCN	0.0853	0.1442	0.0757	0.0956	0.1262	0.1371
Mult-DAE	0.0966	0.1730	0.0803	0.1068	0.1312	0.1448
DimeRec	0.1214	0.2012	0.1043	0.1241	0.1612	0.1811
G-Diff	0.1291	0.2060	0.1124	0.1390	0.1831	0.1951
DiffRec	0.1221	0.2050	0.1055	0.1346	0.1696	0.1823
DiffRec-L	0.0960	0.1681	0.0806	0.1059	0.1321	0.1447
ACDGRec	**0.1317**	**0.2065**	**0.1123**	**0.1412**	**0.1852**	**0.1963**
Imp-G	2.00%	1.92%	-0.08%	1.58%	1.14%	0.61%

Table 2: Scores on the Beauty dataset

Metric	R@10	R@20	N@10	N@20	M@10	M@20
GCL-DT	0.0459	0.0625	0.0265	0.0321	0.0251	0.0235
NGCF	0.0355	0.0595	0.0185	0.0251	0.0150	0.0168
LightGCN	0.0487	0.0700	0.0249	0.0327	0.0200	0.0221
CIGCN	0.0394	0.0628	0.0201	0.0266	0.0165	0.0183
Mult-DAE	0.0477	0.0681	0.0255	0.0333	0.0206	0.0229
DimeRec	0.0432	0.0613	0.0245	0.0307	0.0231	0.0219
G-Diff	0.0481	0.0707	0.0274	0.0337	0.0234	0.0253
DiffRec	0.0392	0.0570	0.0202	0.0251	0.0164	0.0178
DiffRec-L	0.0408	0.0603	0.0219	0.0274	0.0183	0.0198
ACDGRec	**0.0497**	**0.0722**	**0.0286**	**0.0357**	**0.0229**	**0.0249**
Imp-G	3.32%	2.12%	4.38%	5.93%	-2.18%	-1.60%

Table 3: Scores on the Toy dataset

Metric	R@10	R@20	N@10	N@20	M@10	M@20
GCL-DT	0.0301	0.0466	0.0186	0.0221	0.0153	0.0159
NGCF	0.0264	0.0399	0.0134	0.0169	0.0101	0.0112
LightGCN	0.0354	0.0515	0.0189	0.0231	0.0149	0.0162
CIGCN	0.0317	0.0481	0.0181	0.0230	0.0147	0.0157
Mult-DAE	0.0272	0.0402	0.0149	0.0183	0.0117	0.0127
DimeRec	0.0297	0.0452	0.0171	0.0215	0.0145	0.0141
G-Diff	0.0323	0.0472	0.0192	0.0231	0.0165	0.0176
DiffRec	0.0270	0.0406	0.0158	0.0194	0.0130	0.0141
DiffRec-L	0.0279	0.0404	0.0154	0.0188	0.0123	0.0134
ACDGRec	**0.0349**	**0.0506**	**0.0199**	**0.0240**	**0.0170**	**0.0183**
Imp-G	8.04%	7.20%	3.65%	3.90%	3.03%	4.00%

Note: (1) To verify that the performance gains are not due to random fluctuations, we conducted a two-tailed paired t-test between ACDGRec and the best-performing baseline (G-Diff). The results show that our model achieves statistically significant improvements across all datasets ($p < 0.05$).
(2) Imp-G denotes the relative improvement of ACDGRec over the best baseline G-Diff.

1.92% compared to the best-performing baseline model. It is worth noting that ACDGRec exhibits a marginal decrease of 0.08% in NDCG@10 compared to the strongest

baseline G-Diff. We argue that this is a reasonable and expected performance trade-off. MovieLens-1M is a highly dense dataset with abundant user-item interactions, allowing standard diffusion models to easily fit the distribution of popular items. In such information-rich scenarios, our introduced contrastive learning and adversarial training modules act as strong structural regularizers. While this strict regularization might slightly constrain the model's ability to over-fit the extreme top-ranked head items, it is precisely this regularization mechanism that prevents the model from collapsing in sparse data and grants ACDGRec its significant superiority in handling cold-start users and resisting malicious poisoning attacks.

On the Beauty dataset, ACDGRec improves Recall@10 and Recall@20 by 3.32% and 2.12%, respectively, while NDCG@10 and NDCG@20 increase by 4.38% and 5.93%, with a particularly significant boost in NDCG. These results demonstrate the model?s effectiveness in capturing user interests and generating more relevant recommendations in sparse environments.

On the Toys dataset, ACDGRec similarly outperforms other models in both Recall and NDCG metrics, with Recall@10 and Recall@20 improving by 8.04% and 7.20%, respectively, showing the model?s strong adaptability and generalization ability in extremely sparse and cold-start data environments.

By combining the evaluation results across the three datasets, it is clear that ACDGRec achieves stable and significant performance improvements across all major metrics. This demonstrates the overall enhancement in recommendation performance and robustness resulting from the integration of Gaussian diffusion modeling, contrastive learning, and adversarial training.

In highly sparse data environments, ACDGRec still maintains strong recommendation performance, with both Recall and NDCG metrics consistently staying at high levels. This confirms that the model retains robust feature modeling and representation learning capabilities even when user behavior data is scarce and supervisory signals are limited.

These results are attributed to the synergistic optimization of the model's internal components, with the contrastive learning and adversarial training modules playing a key role in handling sparsity and enhancing robustness.

The contrastive learning module enhances the discriminability of user and item representations in the embedding space by constructing positive and negative sample pairs and introducing an InfoNCE-based contrastive loss function. Even in cases of highly sparse interaction data or lack of labeled supervision, this module drives the model to learn more discriminative semantic features, effectively mitigating the representation degradation caused by data sparsity. By generating multi-view augmented samples, this module further increases the diversity of the sample distribution, thereby improving the model's generalization ability.

The adversarial training module introduces perturbed samples into the input space, guiding the model to maintain stable outputs even in the presence of noise, outliers, or low-frequency behaviors in non-ideal scenarios. This significantly enhances the robustness of the overall system. The mechanism is particularly effective in addressing cold-start users and long-tail items, strengthening the model?s ability to distinguish and adapt to complex samples.

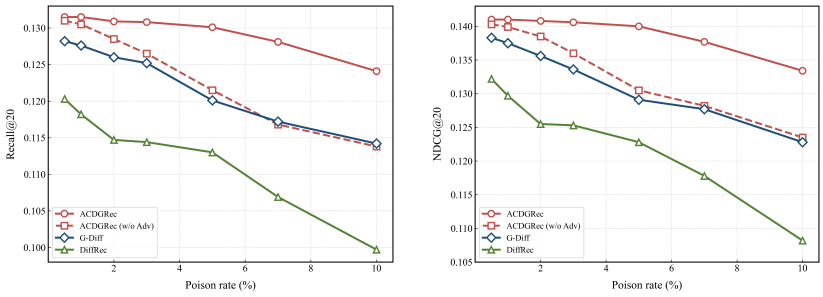


Fig. 4: Performance at ML-1M.

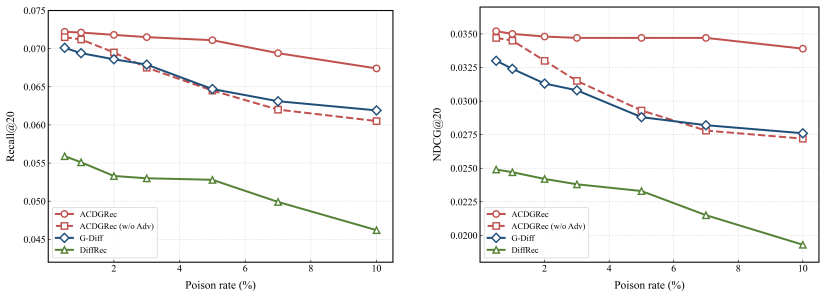


Fig. 5: Performance at Beauty.

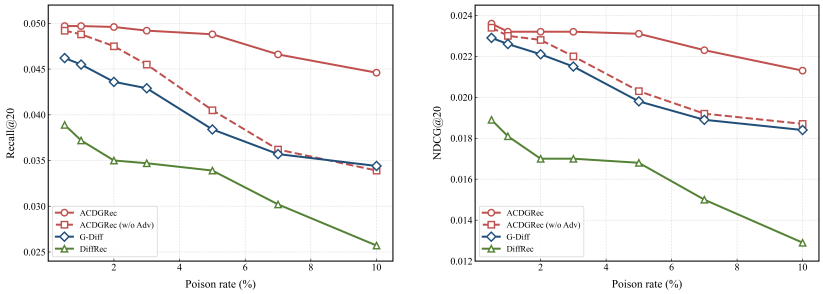


Fig. 6: Performance at Toys.

4.4 Robustness Analysis

To assess the robustness of our method under data-poisoning attacks, we select two representative baselines LightGCN and G-Diff. LightGCN, an efficient graph convolutional model widely used in recommender systems, is effective at handling sparse data and capturing latent user-item relations; G-Diff, a diffusion-based graph recommender, excels at modeling node-level similarity on graphs. Contrasting our approach with these two methods allows us to clearly demonstrate the advantage of the proposed robust training against poisoning attacks.

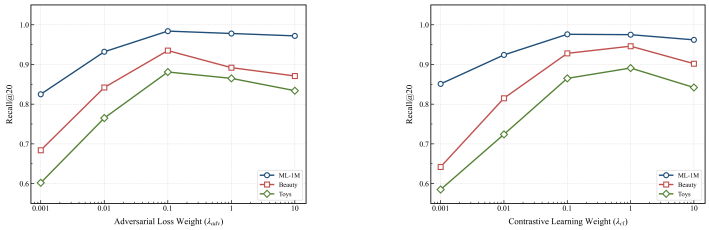


Fig. 7: Performance across Different Numbers of Graph Decoding Layers.

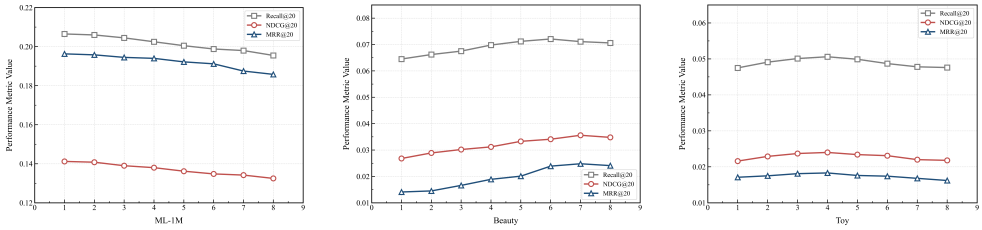


Fig. 8: Performance across Different Numbers of Graph Decoding Layers.

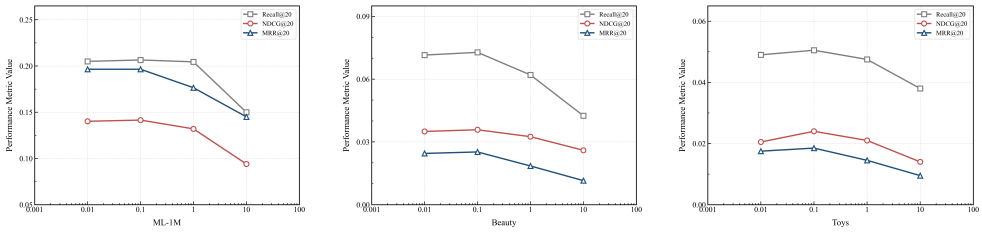


Fig. 9: Performance at Different Noise Scales.

In the poisoning protocol, we inject only fake users and their interactions into the training set. The amount of injected interactions is computed as:

$$M(r) = \lceil \frac{r}{100} \cdot |E_{\text{train}}| \rceil \tag{17}$$

where r is the poison rate, $|E_{\text{train}}|$ the number of observed interactions in the training graph.

The number of interactions per fake user is sampled from the empirical degree distribution of real users, and the interacted items are drawn proportionally to item popularity. We leave the validation split unchanged and do not introduce items that never appear in the clean test set. We vary the poison rate $r \in 0.5, 1, 2, 3, 5, 7, 10\%$. For each rand for each model, we retrain from scratch on the poisoned training set while keeping all other hyperparameters identical to the main experiments; early stopping is determined on the clean validation set. Final performance is reported on the clean test

Table 4: Ablation study of ACDGRec (Metrics at K=20).

Dataset	Model Variant	Recall@20		NDCG@20		MRR@20	
		Score	Drop (↓)	Score	Drop (↓)	Score	Drop (↓)
ML-1M	**ACDGRec (Full)**	**0.2065**	-	**0.1412**	-	**0.1963**	-
	w/o Adv	0.1932	6.44%	0.1283	9.14%	0.1812	7.69%
	w/o CL	0.1212	41.31%	0.0989	29.96%	0.1145	41.67%
Beauty	**ACDGRec (Full)**	**0.0722**	-	**0.0357**	-	**0.0249**	-
	w/o Adv	0.0583	19.25%	0.0246	31.09%	0.0192	22.89%
	w/o CL	0.0341	52.77%	0.0195	45.38%	0.0135	45.78%
Toys	**ACDGRec (Full)**	**0.0506**	-	**0.0240**	-	**0.0183**	-
	w/o Adv	0.0413	18.38%	0.0196	18.33%	0.0142	22.40%
	w/o CL	0.0287	43.28%	0.0119	50.42%	0.0094	48.63%

set using Recall@20 and NDCG@20. The experimental results on the three datasets, including an ablation variant ACDGRec (w/o Adv) to explicitly isolate the defensive contribution, are illustrated in Fig.4, Fig.5, and Fig. 6.

The results show that ACDGRec attains the highest curve envelope and the smallest degradation slope across all poisoning rates and datasets, i.e., at the same contamination level it exhibits lower performance sensitivity. Concretely, ACDGRec performance decreases monotonically with the poison rate but with a markedly flatter decline. Crucially, ACDGRec (w/o Adv) suffers an accelerated performance collapse under severe poisoning, confirming that the adversarial module acts as the primary source of robustness. In contrast, G-Diff typically undergoes an accelerated drop in the mid-high regime, while DiffRec is more brittle in the early-mid regime.

A cross-data-density comparison further indicates that the overall degradation increases with sparsity, yet ACDGRec relative advantage grows in parallel. On Beauty-Sparse, the gap between ACDGRec and the two baselines widens steadily as r increases, suggesting a stronger ability to suppress popularity shift and graph-structure perturbations induced by poisoning. Within the same dataset, the NDCG@20 curve is typically more stable in the low-r region than Recall@20; the two measures diverge further under moderate-to-high poisoning, yet ACDGRec remains state-of-the-art on both metrics and degrades slowest. Overall, across varying data densities and a broad range of poisoning intensities, the proposed method effectively slows performance decay and maintains higher ranking and recall quality, verifying that the adversarial perturbations act as an essential and effective shield against malicious structural noise.

4.5 Parameter Sensitivity Analysis

In this section, we conduct a comprehensive sensitivity analysis on several key hyper-parameters of ACDGRec, including the loss weights (λ_{adv} and λ_{cl}), the number of graph decoding layers (N), and the noise scale, to evaluate the model's stability and robustness.

1) Impact of Loss Weights λ_{adv} and λ_{cl}: As illustrated in Fig.7, we analyze how the adversarial and contrastive learning weights influence performance. The results across three datasets show that ACDGRec maintains stable performance within a wide range of values. Optimal results are generally achieved when $\lambda_{\text{adv}} = 0.1$ and λ_{cl} is around 0.1 to 1.0. The relative flatness of these curves validates that our model is not overly sensitive to loss balancing, which is critical for practical deployment.

2) The number of graph message propagation layers is often crucial in determining GNN performance. As shown in Fig.8, we explore the impact of different numbers of graph decoding layers on recommendation performance and present the following conclusions.

The experimental results show significant differences in the sensitivity of graph neural network layers across various datasets. On the ML-1M dataset, recommendation performance is optimal when the number of graph decoding layers N=1. As the number of layers increases, performance continuously declines, indicating that a shallower structure is more suitable for this dataset. On the Beauty dataset, performance improves with an increasing number of layers. Recall@20 reaches its peak at N=6, and NDCG@20 and MRR@20 reach their best at N=7, suggesting that deeper graph structures are more advantageous for this dataset. On the Toys dataset, performance improves as the number of layers increases, with the best performance achieved at N=4, indicating that the number of layers has a relatively small effect on performance.

3) A key characteristic of diffusion models is the introduction of noise, and the scale of this noise significantly affects performance. Fig.9 illustrates the impact of the noise scale on recommendation performance. The following conclusions can be drawn.

The experimental results across all three datasets show that an inappropriate noise scale?whether too large or too small?can degrade the recommendation performance of ACDGRec. For ML-1M, Beauty, and Toys, the optimal noise scale corresponding to the best recommendation performance is 0.1. This highlights the critical role that the choice of noise scale plays in recommendation effectiveness.

When the noise scale is too large, the model?s recommendations become unstable. Excessive noise can disrupt the model's learning process, making it difficult to converge to an effective solution during training, which in turn reduces recommendation accuracy. Too much noise randomizes the model, hindering its ability to capture the latent relationships between users and items, thereby degrading performance.

Conversely, when the noise scale is too small, the model may overfit the training data. Even if the training data is accurate, a small noise scale may cause the model to focus only on local information, overlooking the diversity and potential variations in the data. This overly conservative learning approach limits the model's ability to explore the recommendation space effectively, restricting its generalization capability and ultimately leading to performance degradation.

Overall, ACDGRec demonstrates high stability across various hyperparameter settings, fulfilling the requirements for robust recommendation in different scenarios.

4.6 Ablation Study

To rigorously verify the contribution of our specific structural designs, we conducted an ablation study focusing on the proposed regularizers. It is worth noting that the

ablation of the foundational backbone, namely isolating the GNN or the diffusion process, is inherently addressed in our overall performance comparison in Section 4.3. Specifically, classical GNN models like LightGCN and NGCF serve as the functional equivalents of a 'w/o Diffusion' variant, while standard diffusion models like DiffRec and DimeRec represent the 'w/o GNN' counterparts. The superiority of the joint Diffusion-GNN architecture, such as G-Diff, over these decoupled models has already been established. Therefore, this section specifically isolates and ablates our novel contributions: the contrastive learning and adversarial training modules. The comparative results of these variants are summarized in Table 4.

As indicated in Table 4, the complete ACDGRec model consistently achieves the best performance across all three datasets, empirically demonstrating the necessity of integrating both modules. Specifically, the variant removing the contrastive learning module (labeled as w/o CL) exhibits the most significant performance degradation on sparse datasets. For instance, on the Beauty and Toys datasets, removing this module leads to a sharp decline in Recall@20, dropping from 0.0722 to 0.0341 and from 0.0506 to 0.0287, respectively. This substantial drop confirms that the contrastive learning mechanism is pivotal for mitigating data sparsity, as it enables the model to learn discriminative representations even when user-item interactions are limited.

Similarly, the variant excluding the adversarial training module (labeled as w/o Adv) also suffers from a consistent performance loss across all metrics. Although the degradation is generally less severe than that of the w/o CL variant on sparse data, it highlights the importance of adversarial perturbations in regularizing the embedding space. By simulating potential noise and attacks during training, the adversarial module effectively enhances the model's robustness and generalization capability, preventing overfitting to the training data. In conclusion, the quantitative analysis in Table 4 validates that the synergistic collaboration between contrastive learning and adversarial training is the key factor driving ACDGRec's superior performance.

The introduction of the MRR@20 metric further confirms the necessity of our proposed modules. As shown in Table 4, removing the contrastive learning module leads to a drastic decline in MRR, indicating that our semantic-aware constraints are vital for accurate item ranking in sparse environments.

Table 5: Performance comparison on cold-start users (interactions < 5).

Model	ML-1M		Beauty		Toys	
	R@20	**N@20**	**R@20**	**N@20**	**R@20**	**N@20**
LightGCN	0.0432	0.0215	0.0124	0.0058	0.0098	0.0042
G-Diff	0.0685	0.0402	0.0186	0.0092	0.0145	0.0071
SDRec	0.0712	0.0428	0.0205	0.0104	0.0162	0.0083
ACDGRec	**0.0784**	**0.0482**	**0.0263**	**0.0141**	**0.0218**	**0.0116**
Improve.	+10.1%	+12.6%	+28.3%	+35.6%	+34.6%	+39.8%

Note: All improvements are statistically significant with $p < 0.05$.

As illustrated in Table 5, the performance of all models drops significantly in the cold-start group compared to the full dataset, reflecting the extreme difficulty of preference modeling under data scarcity. However, ACDGRec consistently maintains a higher performance envelope. Notably, on the Toys dataset, ACDGRec improves Recall@20 by 34.6% and NDCG@20 by 39.8% over SDRec. This is because the graph-aware contrastive learning module provides stable semantic anchors, preventing the diffusion process from generative drift when structural signals are faint.

4.7 Complexity and Efficiency Analysis

Beyond recommendation accuracy, computational efficiency is a critical factor for large-scale deployment. In this section, we analyze the training time and inference latency of ACDGRec in comparison with baseline models like LightGCN, G-Diff, and DimeRec. Table 6 summarizes the results across these configurations.

Table 6: Efficiency comparison on the Beauty dataset.

Model	**Parameters**	**Training (s/epoch)**	**Inference (ms/user)**
LightGCN	2.4M	12.4	0.45
G-Diff	3.1M	45.8	2.12
DimeRec	3.5M	52.3	2.85
ACDGRec	**3.8M**	**58.6**	**3.14**

Table 6 summarizes the complexity comparison. As expected, LightGCN exhibits the lowest latency due to its linear structure. However, compared to the latest generative baselines like DimeRec, our ACDGRec maintains a comparable efficiency profile despite its superior recommendation accuracy. Specifically, the training time of ACDGRec is only slightly higher than DimeRec, which is a reasonable trade-off given the significant performance gains provided by our adversarial and contrastive modules. Most importantly, the inference latency of ACDGRec is approximately 3.14 ms per user. In an industrial context, this is well within the real-time response threshold, proving that ACDGRec is highly suitable for large-scale production environments. Furthermore, the increase in parameter size remains within the memory capacity of standard server GPUs, ensuring the model's scalability.

4.8 Empirical Convergence and Module Coupling Analysis

To address the theoretical concerns regarding the joint optimization of the diffusion process, contrastive learning, and adversarial training, we provide an empirical convergence analysis to verify that these modules operate synergistically without gradient conflicts. Providing a strict theoretical bound for the joint optimization of non-convex deep generative models remains an open mathematical challenge. However, the theoretical design of ACDGRec ensures functional decoupling: the contrastive loss L_{InfoNCE}

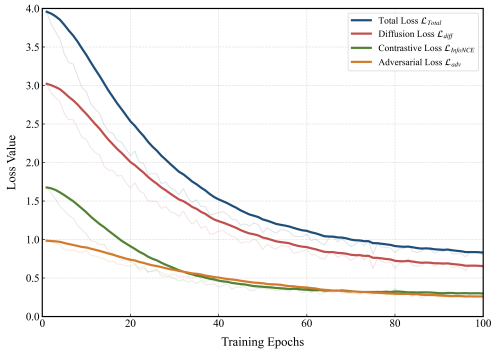


Fig. 10: Empirical convergence curves of the individual loss components and the total joint loss on the Beauty dataset.

acts strictly on the latent representation space prior to the generative process, serving as a structural clustering anchor; meanwhile, the diffusion loss L_{diff} and adversarial loss L_{adv} optimize the reverse denoising trajectory. Because the adversarial perturbations are bounded by a small ϵ-neighborhood as defined in Eq. 8, they serve as local smoothness regularizers rather than disruptive signals, ensuring they do not destroy the original Gaussian generative distribution.

To empirically validate this, we plotted the training loss curves over epochs on the Beauty dataset. As illustrated in Fig.10, the visualization presents the raw batch-level fluctuations depicted by faded lines to reflect authentic training dynamics, alongside the exponential moving average trends represented by bold lines to highlight the macro-convergence behavior. All three individual loss components, specifically L_{diff}, L_{InfoNCE}, and L_{adv}, decrease monotonically and converge simultaneously with the total joint loss L_{Total}. Notably, there are no severe oscillations or divergent spikes among the separate losses. This empirical evidence firmly guarantees that the three objectives are mutually compatible and that the joint optimization converges to a stable and harmonious local optimum.

5 Conclusion

This paper presents ACDGRec, a graph diffusion recommendation model that integrates contrastive learning and adversarial training mechanisms. The model is designed to systematically address core challenges in recommendation systems, such as data sparsity, cold-start problems, and insufficient robustness under noisy data. By introducing a Gaussian diffusion process to model latent relationships between users and items, incorporating graph neural networks to capture complex structural information, leveraging contrastive learning to enhance feature representation, and applying adversarial training to improve robustness and generalization, ACDGRec forms a recommendation framework optimized through the synergistic collaboration of multiple modules.

The experimental results show that ACDGRec outperforms existing strong baseline models on three real-world datasets: MovieLens-1M, Beauty, and Toys. Ablation

experiments further confirm the critical role of contrastive learning and adversarial training in the overall framework, with the contrastive learning module particularly excelling in scenarios with low-frequency interaction data.

In addition, ACDGRec demonstrates strong potential in scalability and module compatibility. Its contrastive learning module and adversarial perturbation mechanism can be flexibly applied to other generative recommendation systems or sequential recommendation scenarios, offering high generalizability and versatility. Future research can expand on several directions. In terms of model architecture, the design of the graph diffusion mechanism can be further deepened to enhance its ability to represent complex structures. Regarding the training mechanism, more advanced learning paradigms can be introduced to improve the model's adaptability and stability in weak supervision or privacy-constrained environments. In terms of application expansion, the model has the potential to further apply to multi-source heterogeneous information processing, addressing more diverse recommendation needs.

References

[1] W. Wang, Y. Xu, F. Feng, *et al.*, "Diffusion recommender model." In: *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, pp. 832–841 (2023). https://doi.org/10.1145/3539618.3591663

[2] P. M. T. Do, Q. Zhang, G. Zhang, J. Lu, "Graph contrastive learning with diffusion-based transfer for cross-domain recommender system." *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 56, no. 1, pp. 375–386 (2025). https://doi.org/10.1109/TSMC.2025.3624447

[3] W. Li, J.-C. Kuo, M. Sheng, *et al.*, "Beyond explicit and implicit: How users provide feedback to shape personalized recommendation content." In: *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Art. no. 893, pp. 1–17 (2025). https://doi.org/10.1145/3706598.3713241

[4] F. A. Croitoru, V. Hondru, R. T. Ionescu, *et al.*, "Diffusion models in vision: A survey." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10850–10869 (2023). https://doi.org/10.1109/TPAMI.2023.3261988

[5] B. B. Moser, A. S. Shanbhag, F. Raue, *et al.*, "Diffusion models, image super-resolution, and everything: A survey." *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 7, pp. 11793–11813 (2024). https://doi.org/10.1109/TNNLS.2024.3476671

[6] D. P. Kingma, M. Welling, "Auto-encoding variational Bayes." *arXiv preprint arXiv:1312.6114* (2014).

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets." In: *Advances in Neural Information Processing Systems*, vol. 27, pp. 2672–2680 (2014).

[8] J. Ho, A. Jain, P. Abbeel, "Denoising diffusion probabilistic models." In: *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 6840–6851 (2020).

[9] P. Mateos, A. Bellogín, "A systematic literature review of recent advances on context-aware recommender systems." *Artif. Intell. Rev.*, vol. 58, no. 1, Art. no. 20 (2024). https://doi.org/10.1007/s10462-024-10939-4

[10] F. Scarselli, M. Gori, A. C. Tsoi, *et al.*, "The graph neural network model." *IEEE Trans. Neural Networks*, vol. 20, no. 1, pp. 61–80 (2008). https://doi.org/10.1109/TNN.2008.2005605

[11] Y. Deldjoo, Z. He, J. McAuley, *et al.*, "A review of modern recommender systems using generative models (gen-recsys)." In: *Proc. 30th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, pp. 6448–6458 (2024). https://doi.org/10.1145/3637528.3671474

[12] E. Nikishin, M. Schwarzer, P. D'Oro, *et al.*, "The primacy bias in deep reinforcement learning." In: *Proc. Int. Conf. Mach. Learn.*, pp. 16828–16847 (2022).

[13] N. Huang, A. Gokaslan, V. Kuleshov, *et al.*, "The GAN is dead; long live the GAN! A modern GAN baseline." In: *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 44177–44215 (2024). https://doi.org/10.52202/079017-1402

[14] M. Gori, G. Monfardini, F. Scarselli, "A new model for learning in graph domains." In: *Proc. 2005 IEEE Int. Joint Conf. Neural Netw.*, vol. 2, pp. 729–734 (2005). https://doi.org/10.1109/IJCNN.2005.1555942

[15] T. N. Kipf, M. Welling, "Semi-supervised classification with graph convolutional networks." *arXiv preprint arXiv:1609.02907* (2016). https://doi.org/10.48550/arXiv.1609.02907

[16] A. Pal, C. Eksombatchai, Y. Zhou, *et al.*, "Pinnersage: Multi-modal user embedding framework for recommendations at Pinterest." In: *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 2311–2320 (2020). https://doi.org/10.1145/3394486.3403280

[17] X. He, K. Deng, X. Wang, *et al.*, "Lightgcn: Simplifying and powering graph convolution network for recommendation." In: *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, pp. 639–648 (2020). https://doi.org/10.1145/3397271.3401063

[18] P. Veličković, G. Cucurull, A. Casanova, *et al.*, "Graph attention networks." *arXiv preprint arXiv:1710.10903* (2017). https://doi.org/10.48550/arXiv.1710.10903

[19] W. Hamilton, Z. Ying, J. Leskovec, "Inductive representation learning on large graphs." In: *Adv. Neural Inf. Process. Syst.*, vol. 30 (2017).

[20] H. Wang, M. Zhao, X. Xie, *et al.*, "Knowledge graph convolutional networks for recommender systems." In: *Proc. World Wide Web Conf.*, pp. 3307–3313 (2019). https://doi.org/10.1145/3308558.3313417

[21] X. Wang, H. He, Y. Cao, *et al.*, "KGAT: Knowledge graph attention network for recommendation." In: *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 950–958 (2019). https://doi.org/10.1145/3292500.3330989

[22] Y. Liu, S. Yang, Y. Xu, *et al.*, "Contextualized graph attention network for recommendation with item knowledge graph." *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 181–195 (2023). https://doi.org/10.1109/TKDE.2021.3082948

[23] J. Walker, T. Zhong, F. Zhang, *et al.*, "Recommendation via collaborative diffusion generative model." In: *Proc. Int. Conf. Knowl. Sci. Eng. Manage.*, pp. 593–605 (2022). https://doi.org/10.1007/978-3-031-10989-8_47

[24] Z. Li, A. Sun, C. Li, "Diffurec: A diffusion model for sequential recommendation." *ACM Trans. Inf. Syst.*, vol. 42, no. 3, pp. 1–28 (2023). https://doi.org/10.1145/3631116

[25] W. Li, R. Huang, H. Zhao, *et al.*, "DimeRec: A unified framework for enhanced sequential recommendation via generative diffusion models." In: *Proc. 18th ACM Int. Conf. Web Search Data Min.*, pp. 726–734 (2025). https://doi.org/10.1145/3701551.3703555

[26] R. Chen, J. Fan, M. Wu, *et al.*, "G-Diff: A graph-based decoding network for diffusion recommender model." *IEEE Trans. Neural Networks Learn. Syst.* (2024). https://doi.org/10.1109/TNNLS.2024.3491827

[27] C. Szegedy, W. Zaremba, I. Sutskever, *et al.*, "Intriguing properties of neural networks." *arXiv preprint arXiv:1312.6199* (2013). https://doi.org/10.48550/arXiv.1312.6199

[28] A. Madry, A. Makelov, L. Schmidt, *et al.*, "Towards deep learning models resistant to adversarial attacks." In: *Proc. Int. Conf. Learn. Represent.* (2018). https://doi.org/10.48550/arXiv.1706.06083

[29] X. He, Z. He, X. Du, *et al.*, "Adversarial personalized ranking for recommendation." In: *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, pp. 355–364 (2018). https://doi.org/10.1145/3209978.3209981

[30] S. Rendle, C. Freudenthaler, Z. Gantner, *et al.*, "BPR: Bayesian personalized ranking from implicit feedback." *arXiv preprint arXiv:1205.2618* (2012). https://doi.org/10.48550/arXiv.1205.2618

[31] Y.-C. Tsai, M. Guan, C.-T. Li, *et al.*, "Predicting new adopters via socially-aware neural graph collaborative filtering." In: *Proc. Int. Conf. Comput. Data Social*

Netw., pp. 155–162 (2019). https://doi.org/10.1007/978-3-030-34980-6_18

[32] R. Hadsell, S. Chopra, Y. LeCun, "Dimensionality reduction by learning an invariant mapping." In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 1735–1742 (2006). https://doi.org/10.1109/CVPR.2006.100

[33] A. Oord, Y. Li, O. Vinyals, "Representation learning with contrastive predictive coding." *arXiv preprint arXiv:1807.03748* (2018). https://doi.org/10.48550/arXiv.1807.03748

[34] J. Wu, X. Wang, F. Feng, *et al.*, "Self-supervised graph learning for recommendation." In: *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, pp. 726–735 (2021). https://doi.org/10.1145/3404835.3462862

[35] Y. Li, Z. Yu, G. He, *et al.*, "Spd-ddpm: Denoising diffusion probabilistic models in the symmetric positive definite space." In: *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 12, pp. 13709–13717 (2024). https://doi.org/10.1609/aaai.v38i12.29276

[36] C. Meng, J. Gao, Y. Tian, *et al.*, "Generative modeling of seismic data using score-based generative models." In: *Proc. 85th EAGE Annu. Conf. Exhib.*, vol. 2024, no. 1, pp. 1–5 (2024). https://doi.org/10.3997/2214-4609.202410861

[37] S. Nie, H. A. Guo, C. Lu, *et al.*, "The blessing of randomness: SDE beats ODE in general diffusion-based image editing." In: *Proc. Int. Conf. Learn. Represent.* (2024).

[38] R. Zhang, Y. Huang, Y. Lou, *et al.*, "Synergistic attention-guided cascaded graph diffusion model for complementarity determining region synthesis." *IEEE Trans. Neural Networks Learn. Syst.*, vol. 36, no. 7, pp. 11875–11886 (2024). https://doi.org/10.1109/TNNLS.2024.3477248

[39] D. Li, Z. Wang, Y. Chen, *et al.*, "A survey on deep active learning: Recent advances and new frontiers." *IEEE Trans. Neural Networks Learn. Syst.*, vol. 36, no. 4, pp. 5879–5899 (2024). https://doi.org/10.1109/TNNLS.2024.3396463

[40] T. Chen, S. Kornblith, M. Norouzi, *et al.*, "A simple framework for contrastive learning of visual representations." In: *Proc. Int. Conf. Mach. Learn.*, pp. 1597–1607 (2020).

[41] Z. Zhuang, M. Liu, A. Cutkosky, *et al.*, "Understanding AdamW through proximal methods and scale-freeness." *Trans. Mach. Learn. Res.* (2022). https://doi.org/10.48550/arXiv.2202.00089

[42] X. Wang, H. He, M. Wang, *et al.*, "Neural graph collaborative filtering." In: *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, pp. 165–174 (2019). https://doi.org/10.1145/3331184.3331267

[43] T. Zhu, L. Sun, G. Chen, "Embedding disentanglement in graph convolutional networks for recommendation." *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 431–442 (2021). https://doi.org/10.1109/TKDE.2021.3087791

[44] Y. Zhang, Y. Zhang, D. Yan, *et al.*, "Revisiting graph-based recommender systems from the perspective of variational auto-encoder." *ACM Trans. Inf. Syst.*, vol. 41, no. 3, pp. 1–28 (2023). https://doi.org/10.1145/3573385