SimDiff: A Simple yet Efficient Diffusion-based Collaborative Filtering Framework

Anonymous Authors

***Abstract*—Diffusion models have demonstrated promising potential in recommender systems owing to their powerful generative ability. However, due to the inherent sparse nature of real-world recommendation data and the inconsistency in the variation of reconstruction and ranking losses during training, existing works suffer two issues: 1) Randomly sampled Gaussian noise addition tends to obscure original user preferences. 2) Training for generation and preference learning tasks interferes with each other, limiting the generative ability of the model. To address these issues, we propose SimDiff, a simple and novel diffusion-based recommendation framework. For the first issue, instead of using random Gaussian noise, we leverage rich semantic information by incorporating auxiliary signals from text or image modalities to enhance the input data of denoising model. In response to the second issue, based on a comprehensive analysis of the mutual influence between generation and preference learning in diffusion recommender systems, we build a collaborative training objective strategy to transform the interference between them into mutual collaboration, which jointly enhances the model training effectiveness. Additionally, we employ multiple GCN layers only during inference to incorporate higher-order co-occurrence information while maintaining training efficiency. Extensive experiments on four real-world datasets demonstrate that SimDiff significantly outperforms state-of-the-art methods. Our SimDiff offers a simple yet effective solution for enhancing recommendation performance and suggests a novel paradigm for applying diffusion method in recommender systems.**

***Index Terms*—Collaborative Filtering, Generative Recommender Model, Diffusion Model.**

I. Introduction

In the age of data explosion, recommender systems have become crucial for managing the exponential growth of information. As the volume of user interaction data continues to grow, there is an increasing demand for recommender systems to effectively extract potential user preferences. In recent years, generative models have attracted considerable attention from the research community due to their impressive ability to model complex data distributions and generate highly realistic outputs [1]–[10]. Among various generative models, diffusion models have emerged as a particularly advantageous paradigm for their exceptional performance in capturing data distributions [11]–[20].

Diffusion-based models have showcased their promising potential in recommendation and achieved some progress. One notable work is DiffRec [21], which applies the diffusion paradigm directly to user-item interaction graphs. This model implements a training process that involves adding and removing noise from the graph. During inference, it treats the original interaction graph as noisy data and performs denoising to generate predictions. In the domain of sequence recommendation, DreamRec [22] proposes a learning-to-generate paradigm that firstly constructs guidance representations, which are then leveraged for generating an oracle item to depict the true preference of the user directly. Recently, DDRM [23] presents a model-agnostic diffusion framework that first employs a backbone model to train representations, then facilitates bidirectional guidance between users and items, while CF-Diff [24] adapts diffusion with a forward process smoothing item-item similarity. Beyond these, other works [25]–[29] explore diffusion techniques and further enrich the landscape of diffusion-based recommendation research.

Despite the progress made by existing diffusion-based recommendation models, several limitations remain. Current methods primarily adopt a straightforward transfer of diffusion paradigm from image synthesis, simply combining the reconstruction and BPR losses to train the generation and preference learning tasks, wherein the original interaction or item representations undergo randomly sampled Gaussian noise corruption. However, due to the inherently highly sparse recommendation data in the real world and the inconsistency in the objectives of reconstruction and ranking losses during training, this paradigm faces two critical issues when applied to recommendation scenarios:

- **The destruction of interaction information by random Gaussian noise:** The key of dealing with user-item interaction data is ensuring the preservation of the valuable information inherent in these interactions. However, when randomly sampled Gaussian noise is directly added into the user-item interaction graph or representations, it introduces perturbations that do not align with the original data structure. Since Gaussian noise is uncorrelated with the actual interactions, it distorts the true relationships between users and items, thus aggravating the sparsity challenge inherent in the raw data.
- **The inconsistency between generation and preference learning objectives harms the generative ability:** The objective of the BPR loss is to uncover users' underlying preferences, whereas the generation process seeks to recover representations to their original states. This divergence in objectives prevents the model from fully leveraging the generative and generalization capabilities of the diffusion paradigm. Existing models overlook this mismatch and simply combine the two objectives together, which hinders the model's ability to effectively align user preferences through generation.

To address the aforementioned challenges, we investigate the diffusion paradigm on recommender systems and make

some novel modifications. **Regarding the first issue**, instead of employing randomly sampled Gaussian noise, we incorporate auxiliary information derived from text and image modalities, which are rich in semantic and contextual information. Specifically, we directly combine the auxiliary information and item representations using weighted aggregation, and feed the result into the generation process for denoising and generation. This not only injects semantic features into sparse interaction data to enrich its information, but also leverages the subsequent denoising process to eliminate the noise contained in these auxiliary signals, ultimately generating representations that capture users' authentic preferences. **As for the second issue**, after conducting thorough investigations and experiments, we propose a collaborative training objective strategy based on normalization techniques to harmonize generation and preference learning. Our intuition is to allow the diffusion generation process and preference ranking to complement each other and optimize jointly, ultimately generating item representations that capture authentic user preferences. **Additionally**, we only utilize multiple GCN layers in the inference phase to further incorporate higher-order co-occurrence information, which eliminates the need for convolution operations during training, thereby significantly improving efficiency.

In this paper, we propose a simple and effective diffusion based model called **SimDiff**, which enhances semantic information of latent variables by injecting auxiliary signals to item representations. To collaborate generation and preference learning, we use a collaborative training objective based on the normalization technique. Multiple GCN layers are only used during inference to capture higher-order co-occurrence, eliminating convolution operations in training and boosting efficiency. Extensive experiments on five datasets verify the superiority of our SimDiff model. Our contributions can be summarized as follows.

- We propose a novel generative framework that substantially modifies the diffusion paradigm to address the sparsity of recommendation data as well as the mutual interference between existing training objectives.
- We introduce an auxiliary signal containing semantic information extracted from various modal features, instead of corrupting interactions with randomly sampled Gaussian noise, thereby enabling the model to excavate authentic user preferences within latent variables enriched with abundant item features.
- Based on extensive investigation and experiments, we build a novel collaborative training objective strategy which transforms the interference between generation and preference learning into mutual collaboration, thereby substantially improving the model's learning ability and adaptability.
- We conduct evaluations on five real-world interaction datasets. Results show that our model significantly outperforms other baseline methods. Apart from this, we also perform signal-to-noise ratio (SNR) analysis, visualization of loss curves variation to illustrate the superiority and interpretability of SimDiff.

II. RELATED WORK

A. Collaborative Filtering

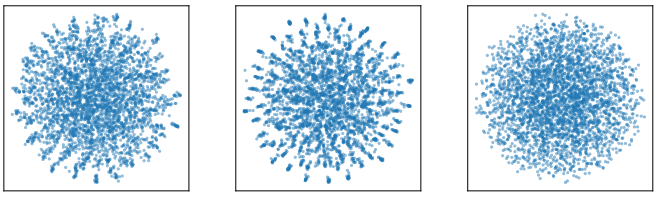
Research in recommender systems began in the 1990s with content-based and collaborative filtering approaches, with a major shift during the Netflix Prize competition that established matrix factorization (MF) techniques [30]–[32] as the dominant models between 2008 and 2016. While MF methods captured preference patterns via latent factors, they struggled with data sparsity and non-linear relationships, leading to the development of Neural Collaborative Filtering (NCF) [33], which utilized deep neural networks for more complex user-item interactions. The field evolved further with Graph Neural Network (GNN) approaches, starting with Neural Graph Collaborative Filtering (NGCF) [34], which encoded collaborative signals through message passing but faced complexity and over-smoothing. A breakthrough came with LightGCN [35], which simplified graph convolution operations by showing that basic neighborhood aggregation, without feature transformation and non-linear activation, effectively captured collaborative signals while reducing computational complexity.

Recent advances in recommender systems focus on contrastive methods for enhanced representation learning. This started with SimCLR [36] in computer vision, adapted for recommendation tasks. Self-supervised Graph Learning (SGL) [37] introduced data augmentation techniques like node dropout, edge dropout, and random walks to create diverse views of the user-item graph. Neighbor Contrastive Learning (NCL) [38] advanced this by using a neighbor-based contrastive objective for more nuanced negative sampling. Recent models like SCCF [39] unify graph convolution with contrastive learning, while RGCL [40] uses adversarial perturbations for a balance between contrastive hardness and rationality. RecDCL [41] combines batch-wise and feature-wise contrastive objectives in a dual framework.

B. Diffusion Based Recommendation

Diffusion models have achieved remarkable success since DDPM [42], which established the foundational framework for learning data distributions through iterative Gaussian noise addition and denoising. Subsequent improvements focused on enhancing sampling efficiency and flexibility. For instance, non-Markovian processes [43] introduced accelerated sampling by redefining the diffusion trajectory, while conditional generation techniques [44] enabled precise control over output characteristics through auxiliary inputs like class labels or textual prompts. These innovations laid the groundwork for adapting diffusion principles to data in recommendation scenarios.

In recommender systems, DiffRec [21] pioneered this adaptation by reformulating the generation of user-item interaction graphs as a denoising process. DreamRec [22] integrated sequential user histories into the diffusion framework, employing time-aware reweighting to emphasize recent interactions and model evolving preferences. Recent advancements explore sophisticated conditioning mechanisms and physical-inspired paradigms. DDRM [23] introduced mutual conditioning between users and items during the reverse diffusion



Pre-trained Noise Signal

Fig. 1: Visualization of the item embeddings on Tiktok dataset using T-SNE.

process, enabling simultaneous refinement of both entities' representations through shared gradient updates. Meanwhile, GiffCF [25] reinterpreted diffusion as a graph heat equation simulation, propagating user-item affinity signals across the interaction graph's Laplacian matrix. These works highlight the potential of diffusion models in modeling complex user-item interactions.

III. Investigation of Diffusion-Based Recommender Systems

A. Comparison between Noise Addition and Auxiliary Signal Injection

In order to investigate the corruption of co-occurrence relationships in recommendation data caused by randomly sampled Gaussian noise, as well as validating the effectiveness of auxiliary semantic signal injection proposed in our SimDiff, we design three kinds of item representations and visualize them using t-SNE for intuitive observation of data distributions. Specifically, we first obtain item representations through LightGCN pre-training on the Tiktok dataset, and then define three variants based on the representations: 1) pre-trained item embeddings that only preserve co-occurrence relationships; 2) representations corrupted by random Gaussian noise; and 3) latent variables obtained through auxiliary signal injection. The second and third variants represent the input data of denoising models in traditional diffusion paradigms and our SimDiff, respectively.

As shown in Figure 1, the item embeddings pre-trained by LightGCN demonstrate a gradual trend toward homogeneous distribution. However, due to the sparsity of original interaction data, this even spread remains limited, with numerous clustered item representations still present. The noise corruption results in items becoming crowded in limited discrete regions of the item space, making them indistinguishable, further intensifying the model's difficulty in capturing inherent user preferences. In stark contrast, after auxiliary signal injection in SimDiff, the embeddings exhibit a more balanced spatial arrangement. This empirical observation strongly suggests that introducing noise to inherently sparse recommendation data significantly disrupts the original interaction patterns, indicating that the forward process of traditional diffusion paradigm is inadequate for handling recommendation scenarios.

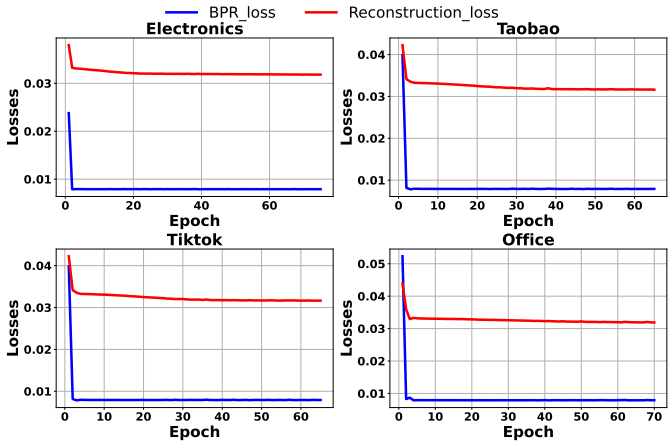


Fig. 2: Comparison between reconstruction and BPR losses in DDRM

B. Comparison of Reconstruction and BPR Losses During Training

To investigate the objective discrepancy between generative modeling and preference learning during the training of diffusion-based models, we collected the reconstruction loss and BPR loss at each training epoch of the DDRM model and visualized them, as shown in Figure 2.

From Figure 2, we can have two observations: (1) There is a substantial gap between the two loss terms, with the reconstruction loss reaching up to ten times that of the BPR loss; (2) Both losses converge within relatively few training iterations. These findings indicate that there is severe mutual interference between the two objectives. The significantly larger magnitude of the reconstruction loss diminishes the influence of the BPR loss during gradient descent. Moreover, the sparsity of the data limits the amount of information that the BPR loss can capture, causing it to converge rapidly. As a result, the representations change less over time, leading to a rapid convergence of the reconstruction loss as well, thereby constraining the model's generative ability. Similar issues can also be observed in the experiments in Section VI-E.

C. Impact of Learning Objectives on Diffusion Model's Generative Ability

In this subsection, we investigate the impact of model's learning ability and generative performance imposed by learning objectives. We select diffusion-based recommender systems CF-Diff, DDRM and DiffRec for comparison with our SimDiff framework on the Tiktok and Taobao dataset. To ensure fairness, we calculate the percentage of changes in generated results for each epoch compared to the previous one, which can be formulated as follows:

$$\mathcal{P}_t = \frac{1}{|\mathcal{N}_r|} \sum_{i=1}^{\mathcal{N}_r} \frac{|r_{i,t} - r_{i,t-1}|}{|r_{i,t-1}|}, \tag{1}$$

where r represents an individual element from the interaction matrix or item embedding, $\mathcal{N}_r$ denotes the total count of elements, and t indicates the current training epoch.

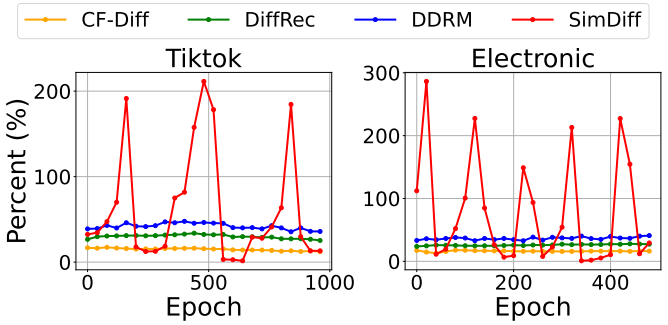


Fig. 3: The percentage of changes in generative outcomes.

It is evident that the percentage of changes for CF-Diff, DiffRec and DDRM remains consistently small and gradually decreases over time, whereas SimDiff maintains a significantly higher level of change throughout. We can clearly observe that the curve of SimDiff exhibits approximately periodic fluctuations, indicating that our framework continuously acquires new information through collaborative training objective strategy. Moreover, the overall performance results in Section VI-B further confirm that SimDiff achieves significantly superior generative performance compared to the other three models. This observation reveals that the interference between generation and preference learning significantly limit the model's continuous learning capabilities, as their training objectives are inconsistent. Conversely, our proposed collaborative training objective strategy effectively transforms the conflict between the two into synergy, enabling continuous learning of the user's true preferences and improving the model's generation performance.

IV. Problem Definition

• **Collaborative Graph with Auxiliary Signal.** Consider the input of a recommender system as a binary interaction graph $\mathcal{G} = (\mathcal{U} \cup \mathcal{I}, \mathcal{E})$, where $\mathcal{U} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_M\}$ represents the set of users and $\mathcal{I} = \{\boldsymbol{i}_1, \boldsymbol{i}_2, ..., \boldsymbol{i}_N\}$ represents the set of items. The edge set $\mathcal{E}$ contains edges between users and items, where an edge $(\boldsymbol{u}_m, \boldsymbol{i}_n) \in \mathcal{E}$ indicates an observed interaction between user $\boldsymbol{u}_m$ and item $\boldsymbol{i}_n$. We can represent the user-item interactions through an adjacency matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, M and N denote the number of users and items. The element $\mathbf{A}_{mn}$ equals $\mathbf{1}$ if there exists an interaction between user $\boldsymbol{u}_m$ and item $\boldsymbol{i}_n$, and $\mathbf{0}$ otherwise. Furthermore, to incorporate rich semantic information to guide the generation process, we introduce the auxiliary signals $\hat{\mathbf{G}}$ extracted from modal features $\hat{\mathbf{F}}$.

• **Task Formulation.** Given this graph, our objective is to learn a function $\boldsymbol{f}$ that predicts the likelihood of future interactions between users and items. For each user $\boldsymbol{u}_m$, we aim to generate a personalized ranking of previously uninteracted items $\{\boldsymbol{i}_n | (\boldsymbol{u}_m, \boldsymbol{i}_n) \notin \mathcal{E}\}$ based on the predicted scores. The function $\boldsymbol{f}$ takes the input of an interaction graph with auxiliary signal $\mathcal{G}^{\mathcal{A}} = (\mathcal{G}, \{\hat{\mathbf{g}}_i | \boldsymbol{i} \in \mathcal{I}\})$, formulated as $\hat{\boldsymbol{y}}_u = f(\mathcal{G}^{\mathcal{A}})$.

V. Methodology

In this section, we present our SimDiff, which consists of training and inference phase. During the training phase, we inject dimensionally-aligned auxiliary information into item representations to enrich their semantic space, treating it as semantically rich noise. After that, we develop a collaborative training objective strategy that continually optimizes the BPR loss while learning the generation process. In the inference phase, after generating item representations, we leverage the LightGCN paradigm to introduce higher-order co-occurrence information, further enhancing the recommendation task performance. We detail each component in the following subsections.

A. Signal Alignment Process

The auxiliary signal, which carries rich semantic information, can be derived from various modalities associated with items, such as user-generated textual reviews, product descriptions, or visual content in the form of item images. Specifically, we first extract the item modal features $\hat{\mathbf{f}}_i \in \mathbb{R}^{d_m}$ by employing different approaches based on the type of modality. For textual data, we utilize a pre-trained Sentence-BERT model as the feature encoder, while for image data, we directly extract the visual features from the raw dataset. Subsequently, to ensure dimensional compatibility and enhance the feature representation, we transform these features through a Multi-Layer Perceptron (MLP) architecture to generate the guide signal $\hat{\mathbf{g}}_i \in \mathbb{R}^d$. This transformation can be formulated as follows:

$$\hat{\mathbf{g}}_i = \text{MLP}(\hat{\mathbf{f}}_i; \theta), \tag{2}$$

where θ represents the learnable parameters of the MLP network, and i denotes the i-th item. This architectural design ensures that the guide signal maintains dimensional consistency with the target space while effectively capturing the essential preference-related information from the input features.

B. Training Phase

In order to better understand personalized user preferences for items and capture latent co-occurrence patterns, we propose a novel representation generation approach. Our key insight is that generating item embeddings directly offers a more comprehensive solution.

*1) **Auxiliary Semantic Signal Injection**:* Considering that user-item interactions typically lack semantic content, we introduce modal signals as auxiliary information and consider them as another form of noise. We synthesize two key information sources: the co-occurrence patterns embedded within user-item interactions and the semantic features extracted from auxiliary signals. Our method combines initialized item embeddings with aligned guide signals through a designed integration process. Specifically, we merge its embedding vector $\mathbf{E}^i$ with the corresponding guide signal $\hat{\mathbf{G}}$ (stacked by $\hat{\mathbf{g}}_i$) through a weighted fusion operation to obtain the latent variable $\mathbf{X}_T$ as follows:

$$\mathbf{X}_T = \mathbf{E}^i * \psi + \hat{\mathbf{G}} * (1 - \psi), \tag{3}$$

where ψ denotes the ratio of combination. This fusion approach preserves both the co-occurrence patterns captured in the item embeddings and the semantic features encoded in the guide signals, while avoiding the potential information loss that would result from noise addition.

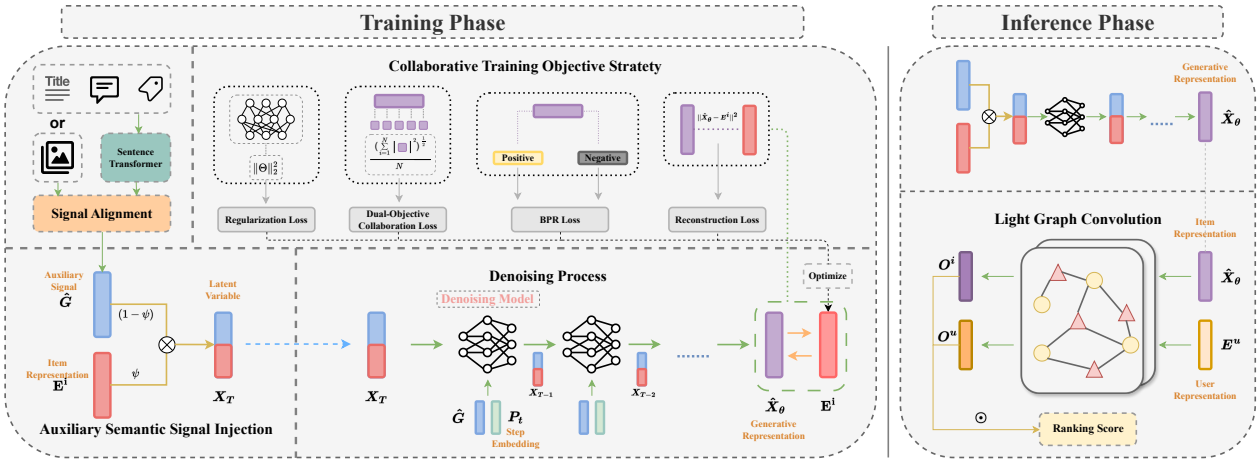


Fig. 4: The overall architecture of our proposed SimDiff, which involves injecting rich semantic information derived from text or image modalities into item representations. Item representations are iteratively updated while guiding the generation process to excavate authentic user preferences. The collaborative training objective strategy continuously optimizes generation and preference learning tasks. During inference, the LightGCN paradigm is incorporated to enhance representations with higher-order co-occurrence information, improving training efficiency by avoiding GCN during the training phase.

*2) **Denoising and Preference Mining Process**:*
Although auxiliary signals in recommender systems contain rich semantic information, not all of them directly reflect authentic user preferences. A substantial portion consists of user preference-irrelevant information that can be treated as noise. In response, we leverage the diffusion reverse paradigm as an effective mechanism to remove such noise while preserving the essential preference signals.

In detail, we initialize the item representations $\mathbf{E}^i$, and employ an MLP structure as denoising model to process latent variables and generate item embeddings. The process is as follows:

$$\mathbf{X}_{T-1} = MLP(Concat[\mathbf{X}_T, \hat{\mathbf{G}}, \mathbf{P}_T]), \tag{4}$$

where $\mathbf{X}_T \in \mathbb{R}^{N\times d}$ is the latent variable, $\hat{\mathbf{G}} \in \mathbb{R}^{N\times d}$ is the guide signal, $\mathbf{P}_T \in \mathbb{R}^{N\times d_t}$ is the time positional encoding at time step T, $\mathbf{X}_{T-1} \in \mathbb{R}^{N\times d}$ is the denoising result of $T-1$ step.

Following the reverse process in existing diffusion paradigm, we finally generate the item embeddings $\hat{\mathbf{X}}_\theta$. Given the parameters θ of model, we define $\mathbf{e}_i$ to denote the recovery target of item i, the t-th learning objective is:

$$L_{t-1} = \sum_{i=1}^{N_I} D_{KL}(q(\mathbf{x}_{i,t-1}|\mathbf{x}_{i,t}, \mathbf{e}_i)||p_\theta(\mathbf{x}_{i,t-1}|\mathbf{x}_{i,t}, \hat{\mathbf{g}}_i)). \tag{5}$$

*3) **Collaborative Training Objective Strategy**:*
The incorporation of auxiliary information enriches the generation process with semantic content. However, since the item representations are trained from initialization, they lack co-occurrence relationships. To integrate user-item interaction patterns while training the generative model, we design a collaborative training objective strategy. Our intuition is to introduce co-occurrence relationships into the generation of item representations. Through the diffusion paradigm, we integrate co-occurrence relationships with semantic information to obtain representations that encapsulate authentic user preferences. In the practical implementation, one of the formulations can be described as:

$$\mathcal{L}_r = \sum_{i=1}^{N} \|\mathbf{e}_i - f_\theta(\mathbf{x}_{i,t}, \hat{\mathbf{g}}_i, \mathbf{P}_t)\|^2. \tag{6}$$

The loss term of reconstruction, denoted as $\mathcal{L}_r$, regulates the evolutionary trajectory of the latent variable $\mathbf{X}_T$ toward the authentic user preference.

We employ the Bayesian Personalized Ranking (BPR) loss as our secondary loss term $\mathcal{L}_{bpr}$. The BPR loss effectively captures pairwise relationships between items, enabling the model to learn from implicit feedback and establish meaningful user-item associations. The BPR loss term $\mathcal{L}_{bpr}$ is described as follows:

$$\mathcal{L}_{bpr} = -\sum_{u=1}^{M} \sum_{i\in N_u} \sum_{j\notin N_u} \ln \sigma(\hat{\boldsymbol{y}}_{ui} - \hat{\boldsymbol{y}}_{uj}). \tag{7}$$

• **Dual-Objective Collaboration.** In the practical implementation, we observe an increasing divergence between reconstruction loss and BPR loss with the training progress, which adversely affects the model's generative capabilities. To further enhance our model's performance and stability, we introduce the dual-objective collaboration loss $\mathcal{L}_c$ that specifically addresses the generation process. This supplementary loss is motivated by a critical observation: there exists a substantial difference between our latent variable x_T and the dynamic target at the beginning of training phase. Without proper constraints and control mechanisms, this discrepancy could potentially lead to unstable and uncontrolled generation. Inspired by the efficiency of regularization loss, we finally adopt the two-paradigm number to constrain the generative outcomes, the loss $\mathcal{L}_c$ can be formally expressed through the following mathematical equation:

$$\mathcal{L}_c = \frac{1}{N}\|\hat{\mathbf{X}}_\theta\|_2 = \frac{1}{N}\left(\sum_i |\mathbf{x}_i^\theta|^2\right)^{1/2}. \tag{8}$$

TABLE I: The comparison of analytical time complexity.

Component	LightGCN	SGL
Adjacency Matrix	$\mathcal{O}(2\|\mathcal{E}\|)$	$\mathcal{O}(4\hat{\rho}\|\mathcal{E}\|s+2\|\mathcal{E}\|)$
Graph Convolution	$\mathcal{O}(2\|\mathcal{E}\|Lds\frac{\|\mathcal{E}\|}{B})$	$\mathcal{O}(2(1+2\hat{\rho})\|\mathcal{E}\|Lds\frac{\|\mathcal{E}\|}{B})$
BPR Loss	$\mathcal{O}(2\|\mathcal{E}\|ds)$	$\mathcal{O}(2\|\mathcal{E}\|ds)$
Self-supervised Loss	-	$\mathcal{O}(\|\mathcal{E}\|d(2+M+N)s)$ / $\mathcal{O}(\|\mathcal{E}\|d(2+2B)s)$
Component	**DiffRec**	**SimDiff**
Forward Process	$\mathcal{O}(BNs)$	$\mathcal{O}(BDds)$
Denoising Process	$\mathcal{O}(kBHNs)$	$\mathcal{O}(kBHds)$
BPR Loss	-	$\mathcal{O}(2\|\mathcal{E}\|ds)$
Reconstruction Loss	$\mathcal{O}(BNs)$	$\mathcal{O}(Bds)$

• **Optimization.** Additionally, we introduce a regularization loss term that serves to constrain the model parameters, preventing overfitting and ensuring stable convergence during the optimization process.The regularization loss term $\mathcal{L}_{reg}$ is:

$$\mathcal{L}_{reg} = \|\Theta\|_2^2. \tag{9}$$

Here, Θ represents the learnable parameters of the model. Taking into account the previously outlined definitions, the consolidated optimization loss used in the training process for recommendation tasks is represented by:

$$\mathcal{L}_{rec} = \alpha_1\mathcal{L}_{bpr} + (1-\alpha_1)\mathcal{L}_r + \alpha_2\mathcal{L}_{reg} + \mathcal{L}_c. \tag{10}$$

Hyperparameters α_1 and α_2 controlling the relative strengths of the ranking and regularization terms.

C. Inference Phase

While the training phase optimizes the model to generate final targets in a single step by leveraging temporal position encoding, the inference phase implements a more fine-grained, step-by-step generation process. Our intuition behind this methodology lies in maximizing the generative potential of the model. By allowing the model to incrementally restructure the information arrangement within the latent variable terms, we achieve two critical objectives: enhanced generation stability and optimal output quality.

Following the generation of item embeddings during the inference phase, we enhance the representation by incorporating higher-order co-occurrence information through the LightGCN paradigm. This facilitates feature propagation between generated item embeddings $\hat{\mathbf{X}}_\theta$ and user embeddings $\mathbf{E}^u$. The process consists of two main steps: First, we process the original interaction graph to obtain its normalized adjacency matrix $\bar{\mathcal{A}}_{u,i}$. Subsequently, the final representations for users $\mathbf{O}^u$ and items $\mathbf{O}^i$ are then obtained through multiple layers of graph convolution operations performed on the normalized adjacency matrix. The formulation is as follows:

$$\mathbf{O}^u = \bar{\mathcal{A}}_{u,*}\mathbf{H}^u,\ \mathbf{O}^i = \bar{\mathcal{A}}_{*,i}\mathbf{H}^i, \tag{11}$$

$$\bar{\mathcal{A}}_{u,i} = \frac{\mathcal{A}_{u,i}}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}. \tag{12}$$

where $\mathbf{H}^u = \mathbf{E}^u,\ \mathbf{O}^u \in \mathbb{R}^{M\times d}$; $\mathbf{H}^i = \hat{\mathbf{X}}_\theta,\ \mathbf{O}^i \in \mathbb{R}^{N\times d}$; $\bar{\mathcal{A}}_{u,i} \in \mathbb{R}^{N\times d}$, $\mathcal{N}_u$ and $\mathcal{N}_i$ denote the neighborhood set of user u and item i in the interaction graph. To obtain the final recommendation predictions, we compute the dot product between the user and item final representations, which produces a recommendation score for each user-item pair. This score quantifies the predicted likelihood of interaction between a given user and item, enabling us to generate personalized recommendations by ranking items.

D. Discussion

*1) **Time Complexity Analysis**:*

In this subsection, we analyze and compare the computational complexity of SimDiff with representative baseline methods including GCN-based LightGCN, contrastive learning-based SGL, and diffusion-based DiffRec. We first define $|\mathcal{E}|$ as the number of edges in the user-item bipartite graph, M and N as the number of users and items. Furthermore, let s denote the number of epochs, B denote the size of each training batch, d denote the embedding size, D denote the embedding size of pre-trained modal feature, L denote the number of GCN layers, k and H denote the layer and hidden size of denoising model, $\hat{\rho} = 1-\rho$ denote the keep probability of SGL. Based on these definitions, we derive the following facts:

- **Training Phase:** We first reduce the dimensionality of the preprocessed modality information using a linear layer, which has a complexity of $\mathcal{O}(BDd)$. Subsequently, the auxiliary signals are added to the item representations to obtain the latent variables. These variables are then processed through an MLP to execute the generation process, with a complexity of $\mathcal{O}(kBHds)$. Given that our collaborative training objective strategy simultaneously optimizes both the BPR loss and the reconstruction loss, their respective complexities are $\mathcal{O}(2|\mathcal{E}|ds)$ and $\mathcal{O}(Nds)$.
- **Inference Phase:** Compared to the training phase, the inference phase involves executing a multi-step denoising process, which results in an additional factor of T being multiplied to the MLP's complexity. Therefore, the overall complexity for the denoising process becomes $\mathcal{O}(TNHd)$. Moreover, since the normalized adjacency matrix has already been generated during the data preprocessing stage, this computation is excluded from the actual model training or testing time.

We summarize the time complexity in training of SimDiff and other methods in Table I. We can clearly observe that SimDiff exhibits marginally higher computational complexity than LightGCN, while being substantially more efficient than both SGL and DiffRec. SGL constructs normalized matrices and performs graph convolution operations in each training iteration and computing self-supervised losses, which significantly increases its computational complexity. DiffRec, on the other hand, necessitates noise injection and denoising operations across all items in each batch during training. By eliminating the noise injection process and due to the fact that the encoding dimension $d << N$, SimDiff achieves notably lower computational complexity compared to DiffRec.

*2) **Interpretability Theoretical Analysis**:*

In this section, we further discuss the positive impact of replacing noise addition with auxiliary semantic signal injection in the model on generative capabilities. Additionally, we analyze

the benefits of the collaborative training objective strategy for model training through gradient analysis.

• **SNR Analysis of the Generation Ability.** In order to further understand the benefits of auxiliary semantic signal injection rather than noise addition in the recommendation scenario, we conduct a theoretical analysis based on signal-to-noise ratio (SNR).

The diffusion reverse process aims to recover x_0 from x_T. In our approach, we inject modality-guided auxiliary signal rather than Gaussian noise, which leads to a higher signal-to-noise ratio (SNR) according to our visualization experiment in Section VI-D. Below we provide a concise proof that high SNR enhances the generative ability of the model. Using the Linear Minimum Mean Square Error Estimator (LMMSE), we model the noisy signal as:

$$x_T = x_0 + m = x_0 + \lambda(\gamma x_0 + \eta) = (1 + \lambda\gamma)x_0 + \lambda\eta, \quad (13)$$

where x_0 is the original signal, η is noise, γ controls alignment between modality noise and original signal, and λ scales the noise. For estimator $\hat{x}_0 = Ax_T$, the optimal coefficient A^* minimizes mean square error:

$$A^* = \frac{(1 + \lambda\gamma)\text{Var}(x_0)}{(1 + \lambda\gamma)^2\text{Var}(x_0) + \lambda^2\text{Var}(\eta)}. \quad (14)$$

As noise variance approaches zero, $A^* \to \frac{1}{1+\lambda\gamma}$, making the recovery:

$$\hat{x}_0 \approx \frac{x_T}{1 + \lambda\gamma}. \quad (15)$$

In recommendation systems, the predicted score is:

$$\hat{r}_{ui} = u^\top \hat{x}_0, \quad \text{where } \hat{x}_0 = f_\theta(x_T). \quad (16)$$

With estimation error δ where $\hat{x}_0 = x_0 + \delta$, the prediction error variance becomes:

$$\text{Var}[\Delta r] = u^\top \text{Cov}[\delta]u, \quad (17)$$

In high-SNR regimes where $\text{Cov}[\delta]$ is small, prediction errors are reduced, leading to more reliable recommendations.

• **Gradient Analysis of the Collaborative Training Objective Strategy.** With this strategy, the gradient of the optimization objective becomes:

$$\nabla_\theta \mathcal{L}_{total} = \nabla_\theta \mathcal{L}_{recon} + \lambda\nabla_\theta \mathcal{L}_c + \beta\nabla_\theta \mathcal{L}_{BPR}. \quad (18)$$

The gradient of the $\mathcal{L}_c$ term is:

$$\nabla_\theta \mathcal{L}_c = \nabla_\theta \|f_\theta(X_T)\|^2 = 2f_\theta(X_T) \cdot \nabla_\theta f_\theta(X_T). \quad (19)$$

When $\|f_\theta(X_T)\|$ is large, the gradient of the $\mathcal{L}_c$ term, $\nabla_\theta \mathcal{L}_c$, will also be large. This increases the penalty on the magnitude of $f_\theta(X_T)$, encouraging the model to generate embeddings with smaller norms.

Let $\alpha = \|f_\theta(X_T)\|$, $\beta = \|X_0\|$, and $\cos\theta$ be the cosine of the angle between the two. The reconstruction loss can be written as:

$$\mathcal{L}_{recon} = \alpha^2 + \beta^2 - 2\alpha\beta\cos\theta. \quad (20)$$

When the term $\mathcal{L}_c = \alpha^2$ is added, the model tends to reduce α. When α is close to β, the reconstruction loss becomes:

$$\mathcal{L}_{recon} \approx 2\beta^2(1 - \cos\theta). \quad (21)$$

TABLE II: Statistics of the datasets

Datasets	Office	Tiktok	Taobao	Electronics
#Users	4,905	9,308	12,539	32,886
#Items	2,420	6,710	8,735	52,974
#Int.	53,258	68,722	83,648	337,837
Sparsity	99.55%	99.88%	99.92%	99.69%
TextDim	768	768	–	300
ImageDim	4096	4096	4096	4096

At this point, the reconstruction loss is mainly determined by the directional difference $\cos\theta$, rather than the norm difference of α and β. This makes the reconstruction objective more consistent with the BPR objective, both focusing on the direction of the representations rather than their magnitudes. This alignment allows the model to simultaneously improve both objectives in the same optimization direction, with the two complementing each other, thereby enhancing the model's ability to accurately capture user preferences.

VI. Experiments

A. Experimental Settings

*1) **Datasets**:*

We conduct experimental evaluations on four widely-used public recommendation datasets: TikTok, Amazon-Office, Amazon-Electronics, and Taobao. The details of each dataset are shown in Table II.

*2) **Evaluation Metrics**:*

The effectiveness of our recommender system was measured using Two standard ranking metrics: **NDCG@*K*** and **Recall@*K***, where ***K*** represents the cutoff threshold for recommended items. We employed the all-rank item evaluation strategy to access accuracy. Final performance metrics were computed by averaging individual scores across all test users.

*3) **Baseline Models**:*

In our experiments, we conduct comprehensive performance comparisons between our proposed framework SimDiff and various existing methods. The baseline models include: (1) classical collaborative filtering methods such as Matrix Factorization (**MF**) [31] and the efficient neural matrix factorization model ENMF [45]; (2) popular GNN-based models including **NGCF** [33] and **LightGCN** [35]; (3) recently proposed contrastive learning-based models that achieve high accuracy, specifically **SGL** [37], **NCL** [38], **SCCF** [39], and **LightGCL** [46]; and (4) state-of-the-art diffusion-based generative models from the past two years, namely **DiffRec** [21], **DDRM** [22], **GiffCF** [25], and **CF-Diff** [24].

*4) **Implementation Details**:*

All models maintain a uniform embedding dimension of 64, and the Xavier initialization method is applied to the embedding parameters. The hyperparameter search space is configured as follows: The learning rate is sampled logarithmically between 1e-6 and 5e-1. For batch size optimization, we select different discrete values based on the interaction volume of each dataset to ensure training efficiency (for instance, choosing a batch size of 1024 for the TikTok dataset and 2000 for the Amazon-Office dataset). The reconstruction alpha

TABLE III: Overall performance comparison between the baselines and SimDiff with Recall@20, Recall@50, NDCG@20, NDCG@50. Bold values indicate the optimal results, while underlined values represent the second-best results. Values marked with * denote statistically significant improvements over the best baseline under single-sample t-test ($p - value < 0.05$). The *%Improv.* illustrates the performance improvement of SimDiff compared to the best baseline model, represented by shaded cells.

Method	TikTok		Office		Taobao		Electronics	
	Recall @20 @50	NDCG @20 @50	Recall @20 @50	NDCG @20 @50	Recall @20 @50	NDCG @20 @50	Recall @20 @50	NDCG @20 @50
MF	0.0557 0.1046	0.0235 0.0332	0.0598 0.1178	0.0232 0.0346	0.0556 0.0983	0.0207 0.0290	0.0401 0.0620	0.0155 0.0198
ENMF	0.1031 0.1656	0.0395 0.0527	0.1004 0.1729	0.0500 0.0651	0.1307 0.1813	0.0630 0.0731	0.0299 0.0512	0.0139 0.0183
NGCF	0.0628 0.1166	0.0245 0.0350	0.0928 0.1684	0.0400 0.0563	0.1223 0.1902	0.0523 0.0658	0.0368 0.0593	0.0163 0.0209
LightGCN	0.0907 0.1471	0.0379 0.0491	0.1215 0.2064	0.0558 0.0702	0.1502 0.2250	0.0681 0.0830	0.0394 0.0645	0.0178 0.0229
SGL	0.0798 0.1308	0.0342 0.0442	0.1151 0.1838	0.0549 0.0697	0.1555 0.2107	0.0748 0.0859	0.0359 0.0561	0.0175 0.0217
NCL	0.0898 0.1447	0.0402 0.0510	0.0966 0.1595	0.0463 0.0594	0.1558 0.2372	0.0717 0.0880	0.0435 0.0679	0.0199 0.0249
LightGCL	0.0911 0.1190	0.0435 0.0455	0.1180 0.1942	0.0531 0.0696	0.1463 0.1986	0.0649 0.0752	0.0379 0.0528	0.0163 0.0208
SCCF	0.0506 0.0883	0.0216 0.0291	0.1221 0.1963	0.0520 0.0644	0.1062 0.1388	0.0540 0.0605	0.0215 0.0332	0.0103 0.0127
DiffRec	0.1036 0.1459	0.0446 0.0536	0.1159 0.1867	0.0511 0.0704	0.1492 0.2013	0.0715 0.0824	0.0236 0.0451	0.0123 0.0189
DDRM-LightGCN	0.0145 0.0218	0.0057 0.0072	0.0133 0.0277	0.0058 0.0088	0.0139 0.0228	0.0057 0.0075	0.0033 0.0044	0.0020 0.0022
DDRM-SGL	0.0281 0.0466	0.0105 0.0147	0.0381 0.0761	0.0156 0.0237	0.0821 0.1086	0.0380 0.0433	0.0060 0.0078	0.0024 0.0028
CF-Diff	0.0665 0.1112	0.0312 0.0402	0.1028 0.1755	0.0500 0.0658	0.0529 0.0731	0.0234 0.0274	0.0099 0.0192	0.0048 0.0067
GiffCF	0.1185 0.1687	0.0462 0.0572	0.1252 0.2084	0.0537 0.0719	0.1524 0.2084	0.0659 0.0786	0.0343 0.0509	0.0138 0.0181
SimDiff	**0.1348*** **0.1885***	**0.0588*** **0.0694***	**0.1361*** **0.2398***	**0.0606*** **0.0808***	**0.1893*** **0.2803***	**0.0783*** **0.0965***	**0.0498*** **0.0763***	**0.0217*** **0.0278***
%Improv.	13.77% 11.73%	27.33% 21.37%	8.75% 15.04%	8.60% 12.38%	21.50% 18.17%	4.68% 9.66%	14.48% 12.37%	9.05% 11.65%

parameter α_1, which controls the strength of the pairwise ranking loss, is searched within the range of 0.5 to 1.0, while the regularization alpha parameter α_2 is explored between 0.001 and 0.01 to find the optimal regularization strength. The number of GCN layers during the inference stage is tested with varying configurations, ranging from 1 to 3 layers. For temporal aspects, we investigate various timestep configurations from 100 to 500. Finally, we compare the performance of two optimizers: Adam and AdamW, both widely recognized for their effectiveness in deep learning applications.

B. Performance Comparison

Table III presents a comparative analysis of our proposed model against various baseline models across four datasets, from which we can have following observations:

- Traditional matrix factorization models decompose user-item interaction matrices to learn latent features but perform poorly by only considering direct interactions, missing higher-order relationships. GCN-based recommender systems like NGCF and LightGCN improve by modeling user-item interactions as bipartite graphs, capturing higher-order connectivity for better representations. However, GCN models may suffer from over-smoothing, making node representations too similar. Contrastive learning alleviates this by creating positive and negative sample pairs, maximizing representation consistency for the same node across different views while preserving node discrimination.
- Diffusion-based recommendation models like DiffRec and GiffCF outperform other baseline methods by modeling complex relationships between user behavior and item features through noise addition and reverse process learning. Their generative nature fosters diversity in recommendations, enabling content discovery. However, the noise from random sampling can disrupt sparse interaction patterns, and static response objectives limit their generative power.
- Our SimDiff outperforms other state-of-the-art models in metrics across all datasets, achieving the best overall performance. This highlights the effectiveness of incorporating auxiliary information to build latent variables, which avoids disruption from Gaussian noise while enriching representations with semantic information. Additionally, the collaborative training objective strategy transforms the mutual interference between generation and preference learning into collaboration, significantly improving generation performance.

C. Ablation Analysis

Table IV presents the ablation study results. In this analysis, **(G, I+G)** denotes the variant where auxiliary signal **G** serves as input data, while the fusion of auxiliary signal

TABLE IV: Ablation analysis results

Method	Metric	TikTok	Office	Taobao	Electronics
(G, I+G)	Recall@20	0.1219	0.1314	0.1456	0.0412
	Recall@50	0.1909	0.2240	0.2341	0.0682
	NDCG@20	0.0520	0.0549	0.0556	0.0183
	NDCG@50	0.0660	0.0741	0.0732	0.0239
(G, I)	Recall@20	0.1204	0.1318	0.1453	0.0426
	Recall@50	0.1903	0.2146	0.2415	0.0691
	NDCG@20	0.0513	0.0518	0.0569	0.0191
	NDCG@50	0.0655	0.0691	0.0760	0.0245
(I, I+G)	Recall@20	0.1206	0.1349	0.1669	0.0410
	Recall@50	0.1909	0.2301	0.2519	0.0688
	NDCG@20	0.0511	0.0553	0.0681	0.0184
	NDCG@50	0.0653	0.0750	0.0850	0.0241
w/o modal	Recall@20	0.0955	0.1249	0.1781	0.0361
	Recall@50	0.1328	0.2135	0.2680	0.0601
	NDCG@20	0.0470	0.0494	0.0732	0.0146
	NDCG@50	0.0542	0.0679	0.0911	0.0195
Pretrain	Recall@20	0.0899	0.1275	0.1775	0.0424
	Recall@50	0.1449	0.2166	0.2581	0.0690
	NDCG@20	0.0361	0.0587	0.0789	0.0185
	NDCG@50	0.0469	0.0771	0.0949	0.0240
SimDiff	Recall@20	**0.1348**	**0.1361**	**0.1893**	**0.0498**
	Recall@50	**0.1885**	**0.2398**	**0.2803**	**0.0763**
	NDCG@20	**0.0588**	**0.0606**	**0.0783**	**0.0217**
	NDCG@50	**0.0694**	**0.0808**	**0.0965**	**0.0278**

and item representation is utilized as the training target for the generative process. The variants **(G, I)** and **(G, I+G)** follow similar patterns. **w/o modal** denotes a variant that does not utilize modality information. Instead, it follows the conventional diffusion paradigm for noise addition. **Pretrain** represents a variant where the generative process target is replaced with pre-trained representations and only train the denoising model.

The results demonstrate that SimDiff achieves superior performance across almost all metrics, validating the efficacy of our proposed paradigm. The variants **(G, I+G)**, **(G, I)**, and **(I, I+G)** achieve competitive secondary results across metrics, indicating their potential viability. These results substantiate the effectiveness of incorporating auxiliary signals as enriched semantic information.

Besides, when modality is not used as an auxiliary semantic signal, its performance on most datasets is only slightly better than that of the **Pretrain** variant. This suggests the effectiveness of injecting modality information into the item representation to enrich preference information, rather than directly adding noise. Notably, when the generative target is set to invariable pre-trained representations, we observe a significant performance degradation. This finding highlights the substantial utility of collaborative training objective strategy in our framework. The empirical evidence strongly supports the advantages of our approach in capturing authentic user preference.

D. SNR Comparison Between Noise Addition and Semantic Signal Injection

In this section, we evaluate the signal-to-noise ratio (SNR) characteristics of two datasets, Taobao and Tiktok. The SNR for each dataset is computed based on a standard statistical definition, where SNR is formulated as the ratio between the square of the mean and the variance of the random variable at each epoch. Specifically, for any random variable X, the SNR is given by:

$$\mathrm{SNR}(X) = \frac{(\mathbb{E}[X])^2}{\mathrm{Var}(X)}, \tag{22}$$

where $\mathbb{E}[X]$ denotes the expectation (mean) of X and $\mathrm{Var}(X)$ represents its variance.

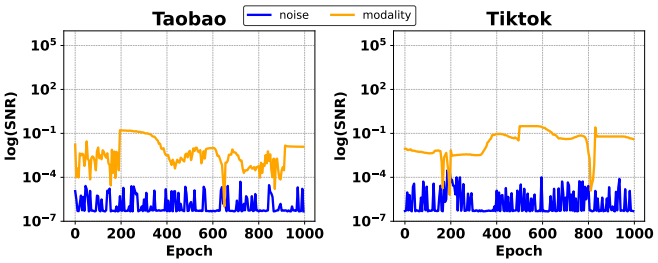


Fig. 5: SNR comparison between noise addition and semantic signal injection

We plot the logarithm of the SNR values ($\log(\mathrm{SNR})$) over training epochs for both datasets, as shown in Figure 5. It is evident that the SNR of latent variables injected with auxiliary semantic signals is significantly higher than that observed in the traditional diffusion paradigm with random noise injection. This indicates that incorporating modality-specific information into item representations substantially enhances the informational content compared to the addition of random noise.

E. Analysis of Collaborative Training Objective Strategy

In this subsection, we demonstrate the core idea of the collaborative training objective (CTO) strategy. We first present a comparison of the results before and after incorporating the dual-objective collaboration loss in Table V. It is evident that the addition of this loss has a substantial impact on the model's performance, significantly enhancing its overall effectiveness.

Same as the existing diffusion models, the reconstruction loss in SimDiff exhibits rapid convergence during the training phase. As discussed in Section V-B3, we implement a collaborative training objective strategy to simultaneously optimize both the generation process and recommendation task objectives. The curves in Figure 6 show that our observations reveal a disparity between the reconstruction loss and BPR loss as training progresses. This divergence becomes so pronounced that the reconstruction loss becomes negligible in comparison to the total loss, halting the continued training of the denoising generative model. The visualization demonstrates the remarkable effectiveness of incorporating the strategy. Before its implementation, the two loss components differed greatly in magnitude. With this loss integrated, the losses balanced to similar values, allowing stable training of the denoising model. This collaborative optimization approach significantly enhances both the generative capabilities and overall model performance, as evidenced by our experimental results.

F. Comparison with Multimodal Baselines

To further evaluate SimDiff's ability to leverage modality information, we compare its performance with several state-of-

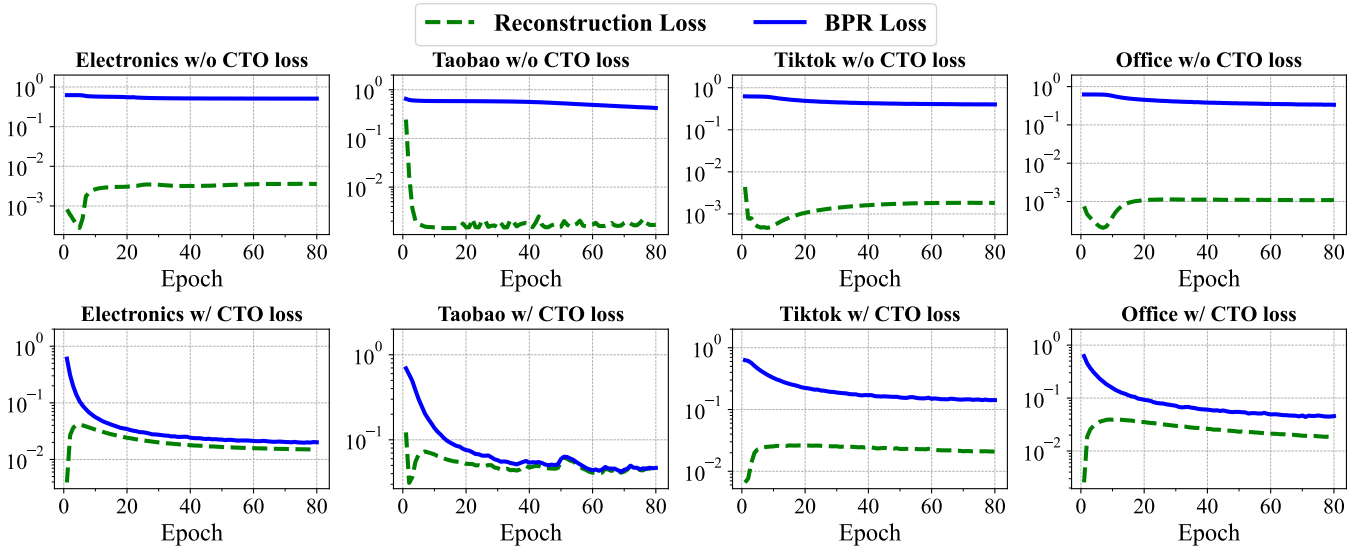


Fig. 6: Comparison of w/o CTO loss and SimDiff

TABLE V: Effectiveness of collaborative training objective strategy

Dataset	@K	w/o CTO		SimDiff	
		Recall	NDCG	Recall	NDCG
TikTok	@20	0.0899	0.0361	**0.1348**	**0.0588**
	@50	0.1449	0.0469	**0.1885**	**0.0694**
Office	@20	0.1275	0.0587	**0.1361**	**0.0606**
	@50	0.2166	0.0771	**0.2398**	**0.0808**
Taobao	@20	0.0592	0.0214	**0.1893**	**0.0783**
	@50	0.1008	0.0297	**0.2803**	**0.0965**
Electronics	@20	0.0022	0.0008	**0.0498**	**0.0217**
	@50	0.0041	0.0012	**0.0763**	**0.0278**

the-art multimodal recommender systems, including **MMSSL** [47], **LATTICE** [48], **BM3** [49], **LGMRec** [50], **MGCN** [51], **SLMRec** [52] and **DiffMM** [53]. While these baseline models incorporate both visual and textual modalities (except for Taobao, which contains only image modality), SimDiff operates using only a single modality. Despite this constraint, SimDiff consistently demonstrates competitive and often superior performance across all four datasets.

These results suggest that SimDiff exhibits a more effective utilization of modality-specific information compared to existing multimodal systems. Notably, on the Taobao dataset—where only a single modality is available—SimDiff significantly outperforms all other methods. This finding indicates that the proposed modality signal injection strategy can effectively enrich the representation of items and capture users' true preferences. It offers a simple yet efficient solution for recommendation tasks, even under limited modality conditions.

G. Indepth Model Analysis

*1) **Cold-start Recommendation**:*

As mentioned in the introduction, data sparsity in recommendation is a critical problem. To prove that the proposed SimDiff has the advantage to solve this issue, we conduct cold-start experiments on Taobao and Amazon-Office datasets, wherein most users have scarce interactions with items. Figure 7 shows the results of cold-start recommendation.

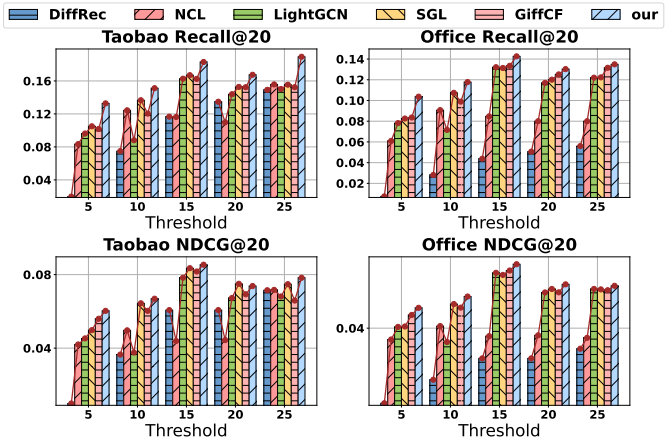


Fig. 7: Performance comparison over Taobao and Amazon-Office between SimDiff and other outstanding baseline models in cold-start recommendation scenario

As illustrated in the figure, the *x-axis* represents different interaction thresholds (5, 10, 15, 20, 25), while the *y-axis* shows the corresponding performance metrics. The visualization demonstrates comparative performance across all methods, with bars representing LightGCN, SGL, NCL, GiffCF, and our proposed framework. SimDiff demonstrates superior performance on sparser interaction data, particularly at lower threshold values of 5, 10, and 15 interactions. Specifically, in the Taobao dataset, it achieves significant improvements in Recall@20 compared to baseline methods, with performance gains of approximately 15%-20% when the interaction threshold is set at these lower values. Similarly, in the Office dataset, we observe even more substantial improvements, with Recall@20 increasing by roughly 20%-40% under the same

TABLE VI: Performance comparison between multimodal recommender systems and SimDiff

Datasets	Metric	MMSSL	LATTICE	BM3	LGMRec	MGCN	SLMRec	DiffMM	SimDiff
TikTok	Recall@20	0.0921	0.0888	0.0988	0.0672	0.1023	0.0967	0.1129	**0.1348**
	Recall@50	0.1513	0.1465	0.1546	0.1053	0.1605	0.1517	0.1810	**0.1885**
	NDCG@20	0.0394	0.0386	0.0399	0.0265	0.0367	0.0346	0.0456	**0.0588**
	NDCG@50	0.0511	0.0501	0.0516	0.0339	0.0485	0.0461	0.0585	**0.0694**
Office	Recall@20	0.1277	0.1345	0.1158	0.1348	0.1196	0.1126	0.1351	**0.1361**
	Recall@50	0.2123	0.2200	0.1944	0.2231	0.2029	0.1915	0.2308	**0.2398**
	NDCG@20	0.0541	0.0524	0.0527	0.0598	0.0544	0.0508	0.0599	**0.0606**
	NDCG@50	0.0732	0.0742	0.0695	0.0789	0.0724	0.0681	0.0804	**0.0808**
Taobao	Recall@20	0.1619	0.1622	0.1451	0.1661	0.1528	0.1474	0.1498	**0.1893**
	Recall@50	0.2377	0.2434	0.2246	0.2392	0.2411	0.2305	0.2342	**0.2803**
	NDCG@20	0.0749	0.0699	0.0636	0.0693	0.0645	0.0624	0.0649	**0.0783**
	NDCG@50	0.0901	0.0862	0.0802	0.0868	0.0829	0.0791	0.0817	**0.0965**
Electronics	Recall@20	0.0425	0.0461	0.0451	0.0449	0.0466	0.0443	0.0467	**0.0498**
	Recall@50	0.0671	0.0712	0.0738	0.0733	0.0756	0.0728	0.0754	**0.0763**
	NDCG@20	0.0214	0.0206	0.0207	0.0209	0.0212	0.0212	0.0215	**0.0217**
	NDCG@50	0.0273	0.0264	0.0267	0.0268	0.0274	0.0271	0.0277	**0.0278**

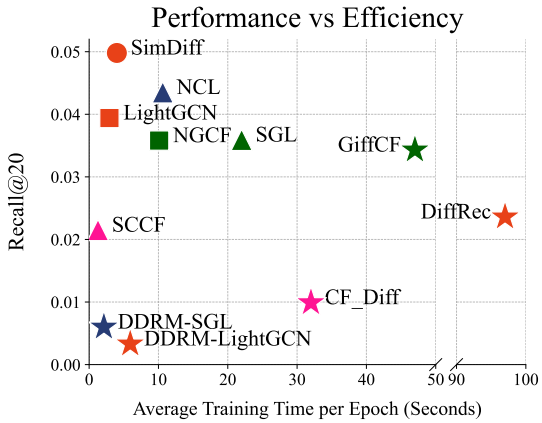


Fig. 8: Performance versus efficiency analysis on Amazon-Electronics. Performance strength and training efficiency increase towards the upper left direction.

sparse interaction conditions. These consistent performance improvements across different domains and sparsity levels provide compelling evidence of our model's strong advantage in handling scenarios with limited user-item interactions.

*2) **Training Efficiency**:*

In this subsection, we aim to study the trade-off between performance and training efficiency. We conduct a performance versus efficiency analysis comparing different models on the Amazon-Electronics dataset which has the most interactions, measuring both the training time per epoch and the Recall@20 metric. To ensure reliability and consistency, all models are evaluated using the same GPU with single-process execution. As illustrated in Figure 8, our SimDiff achieves an optimal balance between training efficiency and model performance, demonstrating superior results while maintaining relatively low training times. Early approaches, such as ENMF, while computationally efficient with shorter training times due to their lower complexity, show poor performance. LightGCN, through its simplified graph convolution operations, maintained high training efficiency and strong performance across most baselines. The contrastive learning paradigm, as demonstrated by NCL, further reinforced its effectiveness in recommendation tasks, achieving second-best performance with acceptable training durations.

TABLE VII: Auxiliary signal analysis

Signal	Metric	TikTok	Office	Electronics
G = Image	Recall@20	0.1310	0.1327	0.0469
	Recall@50	0.1933	0.2246	0.0768
	NDCG@20	0.0588	0.0540	0.0209
	NDCG@50	0.0713	0.0731	0.0271
G = Text	Recall@20	0.1348	0.1361	0.0498
	Recall@50	0.1885	0.2398	0.0763
	NDCG@20	0.0588	0.0606	0.0217
	NDCG@50	0.0694	0.0808	0.0278

*3) **Auxiliary Signal Analysis**:*

In our results of Section VI-B, the performance of the proposed SimDiff is derived from utilizing textual features as auxiliary signals across all datasets except Taobao, which exclusively contains additional image features. To further investigate whether different modalities serving as auxiliary signals influence the model's generative performance, we conducted experiments on the other three datasets that possess both textual and image features. To ensure experimental reliability, we maintained identical hyperparameter settings as those used in the text-based auxiliary signal experiments. The results are shown in Table VII.

As demonstrated by the empirical results, the performance metrics exhibit comparable values across both modalities when utilized as auxiliary signals, with certain metrics under G=Image even surpassing those obtained with the text modality. This observation provides strong evidence that our proposed SimDiff framework effectively leverages rich semantic information across modalities for representation learning and recommendation, with its performance primarily dependent on the semantic richness of auxiliary signals rather than their specific modality type.

H. Hyperparameter Analysis

To investigate the impact of key hyperparameters, we conduct experiments on four benchmark datasets: TikTok, Office, Taobao, and Electronics. The studied hyperparameters include: (1) the number of GCN layers used in the inference phase (GCN_Layers); (2) the reconstruction loss coefficient α_1, with the complementary BPR loss coefficient being $(1-\alpha_1)$; and (3)

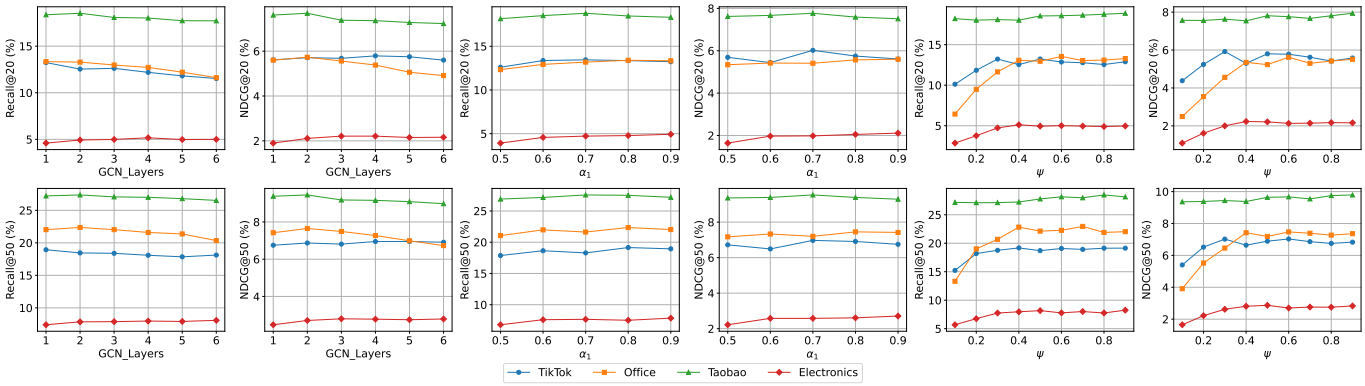


Fig. 9: Hyperparameter sensitivity analysis on four datasets. Rows correspond to different metrics: Recall@20/50 and NDCG@20/50. Columns show performance trends across GCN layers, diffusion loss coefficient α_1, and semantic fusion ratio ψ.

the semantic injection ratio ψ, which controls the proportion of item representation versus semantic signal (with $(1-\psi)$ for the semantic part). The results are shown in Figure 9.

We can observe that increasing GCN layers generally leads to slight performance decline due to over-smoothing, with the best results achieved using 2 or 3 layers. Varying α_1 balances diffusion and BPR losses, with stable performance in the range of $[0.6, 0.8]$, and the most consistent results at $\alpha_1 = 0.7$. As ψ increases, item representations become more dominant over semantic signals, improving performance up to around 0.6 or 0.7, especially for sparse datasets like Electronics and Office.

VII. Conclusion

In this work, we propose a novel diffusion framework called SimDiff for recommender systems. We replace the randomly sampled Gaussian noise addition by injecting auxiliary signal derived from modal features to representations, which introduces rich semantic information to sparse data. In order to improve the generative effect, we build a collaborative training objective strategy which harmonizes the generation and preference learning. Our empirical evaluations across five real-world datasets show that SimDiff significantly outperforms previous diffusion methods. This work presents a novel perspective on diffusion-based recommender systems and suggests new research directions for applying the diffusion paradigm to inherently sparse recommendation tasks.

References

[1] X. Liu, H. Xue, K. Luo, P. Tan, and L. Yi, "Genn2n: Generative nerf2nerf translation," in *CVPR*, 2024, pp. 5105–5114.

[2] C. Jiang, Z. Chen, B. Zhang, Y. Ren, X. Dong, L. Cheng, X. Yang, L. Li, J. Zhou, and L. Mo, "Gats: Generative audience targeting system for online advertising," in *SIGIR*. ACM, 2024, pp. 2920–2924.

[3] L. Sun, J. Hu, S. Zhou, Z. Huang, J. Ye, H. Peng, Z. Yu, and P. Yu, "Riccinet: Deep clustering via a riemannian generative model," in *Proceedings of the ACM on Web Conference 2024*. ACM, 2024, pp. 4071–4082.

[4] S. Gao, J. Fang, Q. Tu, Z. Yao, Z. Chen, P. Ren, and Z. Ren, "Generative news recommendation," in *Proceedings of the ACM on Web Conference 2024*. ACM, 2024, pp. 3444–3453.

[5] X. Wen, H. Zhang, S. Zheng, W. Xu, and J. Bian, "From supervised to generative: A novel paradigm for tabular deep learning with large language models," in *SIGKDD*. ACM, 2024, pp. 3323–3333.

[6] Z. Liu, J. Yang, M. Cheng, Y. Luo, and Z. Li, "Generative pretrained hierarchical transformer for time series forecasting," in *SIGKDD*. ACM, 2024, pp. 2003–2013.

[7] Y. Wang, J. Xun, M. Hong, J. Zhu, T. Jin, W. Lin, H. Li, L. Li, Y. Xia, Z. Zhao *et al.*, "Eager: Two-stream generative recommender with behavior-semantic collaboration," in *SIGKDD*. ACM, 2024, pp. 3245–3254.

[8] Y. Yang, Z. Wu, L. Wu, K. Zhang, R. Hong, Z. Zhang, J. Zhou, and M. Wang, "Generative-contrastive graph learning for recommendation," in *SIGIR*. ACM, 2023, pp. 1117–1126.

[9] D. Lin, L. Jing, X. Song, M. Liu, T. Sun, and L. Nie, "Adapting generative pretrained language model for open-domain multimodal sentence summarization," in *SIGIR*. ACM, 2023, pp. 195–204.

[10] J. Chen, R. Zhang, J. Guo, M. de Rijke, Y. Liu, Y. Fan, and X. Cheng, "A unified generative retriever for knowledge-intensive language tasks via prompt learning," in *SIGIR*. ACM, 2023, pp. 1448–1457.

[11] S. Xue, Z. Liu, F. Chen, S. Zhang, T. Hu, E. Xie, and Z. Li, "Accelerating diffusion sampling with optimized time steps," in *CVPR*, 2024, pp. 8292–8301.

[12] M. Li, T. Cai, J. Cao, Q. Zhang, H. Cai, J. Bai, Y. Jia, M.-Y. Liu, K. Li, and S. Han, "Distrifusion: Distributed parallel inference for high-resolution diffusion models," in *CVPR*, 2024, pp. 7183–7193.

[13] L. Yang, H. Qian, Z. Zhang, J. Liu, and B. Cui, "Structure-guided adversarial training of diffusion models," in *CVPR*, 2024, pp. 7256–7266.

[14] Y. Xu, W. Wang, F. Feng, Y. Ma, J. Zhang, and X. He, "Diffusion models for generative outfit recommendation," in *SIGIR*. ACM, 2024, pp. 1350–1359.

[15] T. Zhong, J. Zhang, Z. Cheng, F. Zhou, and X. Chen, "Information diffusion prediction via cascade-retrieved in-context learning," in *SIGIR*. ACM, 2024, pp. 2472–2476.

[16] X. Long, L. Zhuang, A. Li, H. Li, and S. Wang, "Fact embedding through diffusion model for knowledge graph completion," in *Proceedings of the ACM on Web Conference 2024*. ACM, 2024, pp. 2020–2029.

[17] T.-K. Nguyen and Y. Fang, "Diffusion-based negative sampling on graphs for link prediction," in *Proceedings of the ACM on Web Conference 2024*. ACM, 2024, pp. 948–958.

[18] J. Long, G. Ye, T. Chen, Y. Wang, M. Wang, and H. Yin, "Diffusion-based cloud-edge-device collaborative learning for next poi recommendations," in *SIGKDD*. ACM, 2024, pp. 2026–2036.

[19] H. Zeng, J. Wang, A. Das, J. He, K. Han, H. Hu, and M. Sun, "Effective generation of feasible solutions for integer programming via guided diffusion," in *SIGKDD*. ACM, 2024, pp. 4107–4118.

[20] Y. Jiang, Y. Yang, L. Xia, and C. Huang, "Diffkg: Knowledge graph diffusion model for recommendation," in *WSDM*. ACM, 2024, pp. 313–321.

[21] W. Wang, Y. Xu, F. Feng, X. Lin, X. He, and T.-S. Chua, "Diffusion recommender model," in *SIGIR*. ACM, 2023, pp. 832–841.

[22] Z. Yang, J. Wu, Z. Wang, X. Wang, Y. Yuan, and X. He, "Generate what you prefer: Reshaping sequential recommendation via guided diffusion," *NeurIPS*, vol. 36, 2024.

[23] J. Zhao, W. Wenjie, Y. Xu, T. Sun, F. Feng, and T.-S. Chua, "Denoising diffusion recommender model," in *SIGIR*. ACM, 2024, pp. 1370–1379.

[24] Y. Hou, J.-D. Park, and W.-Y. Shin, "Collaborative filtering based on diffusion models: Unveiling the potential of high-order connectivity," in *SIGIR*. ACM, 2024, pp. 1360–1369.
[25] Y. Zhu, C. Wang, Q. Zhang, and H. Xiong, "Graph signal diffusion model for collaborative filtering," in *SIGIR*. ACM, 2024, pp. 1380–1390.
[26] Z. Yi, X. Wang, and I. Ounis, "A directional diffusion graph transformer for recommendation," in *SIGIR*. ACM, 2024.
[27] Y. Wang, Z. Liu, Y. Wang, X. Zhao, B. Chen, H. Guo, and R. Tang, "Diff-msr: A diffusion model enhanced paradigm for cold-start multi-scenario recommendation," in *WSDM*. ACM, 2024, pp. 779–787.
[28] H. Ma, Y. Yang, L. Meng, R. Xie, and X. Meng, "Multimodal conditioned diffusion model for recommendation," in *Companion Proceedings of the ACM on Web Conference 2024*. ACM, 2024, pp. 1733–1740.
[29] H. Ma, R. Xie, L. Meng, Y. Yang, X. Sun, and Z. Kang, "Seedrec: sememe-based diffusion for sequential recommendation," in *Proceedings of IJCAI*, 2024, pp. 1–9.
[30] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *SIGKDD*. Association for Computing Machinery, 2008, p. 426–434.
[31] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, pp. 30–37, 2009.
[32] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*. AUAI Press, 2009.
[33] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*. ACM, 2017, pp. 173–182.
[34] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *SIGIR*. ACM, 2019, pp. 165–174.
[35] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *SIGIR*. ACM, 2020, pp. 639–648.
[36] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*. PMLR, 2020, pp. 1597–1607.
[37] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *SIGIR*. ACM, 2021, pp. 726–735.
[38] Z. Lin, C. Tian, Y. Hou, and W. X. Zhao, "Improving graph collaborative filtering with neighborhood-enriched contrastive learning," in *Proceedings of the ACM web conference 2022*. ACM, 2022, pp. 2320–2329.
[39] Y. Wu, L. Zhang, F. Mo, T. Zhu, W. Ma, and J.-Y. Nie, "Unifying graph convolution and contrastive learning in collaborative filtering," in *SIGKDD*. ACM, 2024, pp. 3425–3436.
[40] J. Tang, S. Dai, Z. Sun, X. Chen, J. Xu, W. Yu, L. Hu, P. Jiang, and H. Li, "Towards robust recommendation via decision boundary-aware graph contrastive learning," in *SIGKDD*. ACM, 2024, pp. 2854–2865.
[41] D. Zhang, Y. Geng, W. Gong, Z. Qi, Z. Chen, X. Tang, Y. Shan, Y. Dong, and J. Tang, "Recdcl: Dual contrastive learning for recommendation," in *Proceedings of the ACM on Web Conference 2024*. ACM, 2024, pp. 3655–3666.
[42] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NeurIPS*, vol. 33, pp. 6840–6851, 2020.
[43] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *ICML*, 2021.
[44] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," in *NeurIPS*, vol. 34. Curran Associates, Inc., 2021, pp. 8780–8794.
[45] C. Chen, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Efficient neural matrix factorization without sampling for recommendation," *TOIS*, vol. 38, no. 2, pp. 1–28, 2020.
[46] X. Cai, C. Huang, L. Xia, and X. Ren, "Lightgcl: Simple yet effective graph contrastive learning for recommendation," in *ICLR*, 2023.
[47] W. Wei, C. Huang, L. Xia, and C. Zhang, "Multi-modal self-supervised learning for recommendation," in *Proceedings of the ACM Web Conference 2023*. ACM, 2023, pp. 790–800.
[48] J. Zhang, Y. Zhu, Q. Liu, S. Wu, S. Wang, and L. Wang, "Mining latent structures for multimedia recommendation," in *Proceedings of the 29th ACM International Conference on Multimedia*. ACM, 2021, p. 3872–3880.
[49] X. Zhou, H. Zhou, Y. Liu, Z. Zeng, C. Miao, P. Wang, Y. You, and F. Jiang, "Bootstrap latent representations for multi-modal recommendation," in *Proceedings of the ACM Web Conference 2023*. ACM, 2023, p. 845–854.
[50] Z. Guo, J. Li, G. Li, C. Wang, S. Shi, and B. Ruan, "Lgmrec: Local and global graph learning for multimodal recommendation," in *Proceedings of AAAI 2024*. ACM, 2024, pp. 8454–8462.
[51] P. Yu, Z. Tan, G. Lu, and B.-K. Bao, "Multi-view graph convolutional network for multimedia recommendation," in *Proceedings of the 31st ACM International Conference on Multimedia*. ACM, 2023, p. 6576–6585.
[52] Z. Tao, X. Liu, Y. Xia, X. Wang, L. Yang, X. Huang, and T.-S. Chua, "Self-supervised learning for multimedia recommendation," *IEEE Transactions on Multimedia*, 2022.
[53] Y. Jiang, L. Xia, W. Wei, D. Luo, K. Lin, and C. Huang, "Diffmm: Multi-modal diffusion model for recommendation," in *Proceedings of the 32nd ACM International Conference on Multimedia*. ACM, 2024, p. 7591–7599.