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

Anonymous Author(s)

Abstract

Diffusion models have demonstrated promising potential in recommender systems owing to its powerful generative ability. However, due to the inherent sparse nature of real-world recommendation data, existing works suffer two issues: 1) Randomly sampled Gaussian noise addition tends to obscure original user preferences. 2) Relying on static recovery targets with insufficient interaction patterns constrains the model's learning effectiveness and generative ability. 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 leverages 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, we build a dynamic learning target that iteratively updates throughout the training process, enabling richer information capture. A dual-objective collaborative training strategy is designed to simultaneously optimize reconstruction and BPR losses, which coordinated by a dual-objective balance term. Additionally, we employ multiple GCN layers only during inference to incorporate higher-order co-occurrence information while maintaining training efficiency. Extensive experiments on five 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 system.

CCS Concepts

• **Information systems → Recommender systems**.

Keywords

Collaborative Filtering, Generative Recommender Model, Diffusion Model

1 Introduction

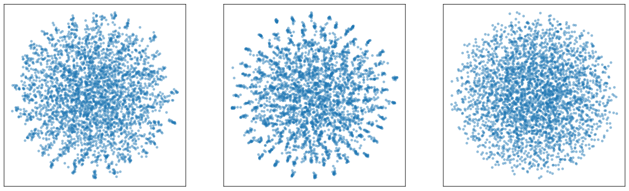
In the age of data explosion, recommender systems have become crucial for managing the exponential growth of information. As the volume of user interaction data continues to grow, there is an increasing demand for recommender systems to effectively extract potential user preferences. In recent years, 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 [3, 6, 11, 16, 18, 19, 27, 32, 33, 39]. Among various generative models, diffusion models have emerged as a particularly advantageous paradigm for their exceptional performance in capturing data distributions [12, 15, 20, 21, 24, 36–38, 42, 45].

Diffusion-based models have showcased their promising potential in recommendation and achieved some progress. One notable work is DiffRec [29], 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 [40] propose 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 [44] 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 [10] adapted diffusion with a forward process smoothing item-item similarity. Beyond these, other works [22, 23, 31, 41, 46] have explored diffusion techniques and further enriching 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, using input data as the recovery target in the denoising model to generate oracle interaction terms, wherein the original interaction or item representations undergo randomly sampled Gaussian noise corruption. However, due to the inherent highly sparse recommendation data in real world, this paradigm faces two critical issues when applied to recommendation scenarios:

- **The destruction of interaction information by 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 aggravate the sparsity challenge inherent in the raw data.
- **Limited available information restricts the generative ability:** During model training, the information density of labels significantly impacts model performance. More informative labels enable models to extract deeper insights of data during training, leading to superior model capability. In the diffusion reverse stage, the recovery targets used during training phase are typically derived from pre-trained representations or existing interaction graphs which contain highly limited user preference pattern. Relying solely on such information sources significantly constrains the generative ability of the model, leading to unreliable generation outcomes.

To address the aforementioned challenges, we investigate the diffusion paradigm on recommendation 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



(a) Pre-trained (b) Noise Addition (c) Signal Injection

Figure 1: Visualization of the item embeddings on Amazon-Baby dataset using T-SNE

semantic and contextual information. 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, we carefully design an dynamic learning target for generative process. Our intuition is to replace the stastic input data with more informative changeable targets updated throughout model training and encompasses increasingly richer information with each iteration. Furthermore, this dynamic target can better leverage the generative ability of the diffusion process, thereby improving the overall generative quality.

In this paper, we propose a simple and effective diffusion based model which called SimDiff. Specifically, in order to introduce semantic information, instead of gradually adding Gaussian noise, we directly combine the auxiliary information and item representations using weighted aggregation, and feed the result into the reverse process for denoising and generation. To construct the dynamic learnable target, we initialize and dynamically update it by using the learning results from the previous epoch as the training target for the next. We also build a dual-objective collaborative training strategy that optimizes both reconstruction loss and BPR loss simultaneously to enhance the renewal of this target. 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. We conduct extensive experiments on five real-world datasets and verify the superiority of our SimDiff model. Our contributions can be summarized as follows.

- We investigate the limitations of previous diffusion-based recommendation models and propose a novel generative framework, in which we significantly modify the diffusion paradigm in response to the sparse nature of recommendation data.
- Instead of adding randomly sampled Gaussian noise to corrupt interactions, we introduce an auxiliary signal with semantic information derived from various modal features, such as text and images, to help generate item representations.
- We build a dynamic target which can be updated iteratively to guide the generation process, thus allowing the model to learn more abundant user preference patterns and substantially enhancing the model's learning capability 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 empirical studies to improve the interpretability of our framework.

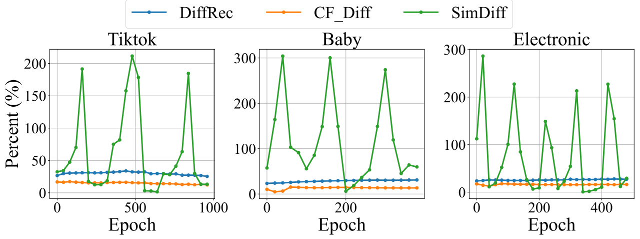


Figure 2: The percentage of changes in generative outcomes

2 Investigation of Diffusion-Based Recommender Systems

2.1 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 Amazon-Baby 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(a), 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. In Figure 1(b), 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, Figure 1(c) clearly shows that 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.

2.2 Impact of Recovery Target on Diffusion Model's Generative Ability

As the recovery target in the diffusion's reverse process guides the denoising trajectory, it largely determines the generative capability of diffusion models. In this subsection, we investigate the restriction of model's learning ability and generative performance imposed by static recovery targets, which extracted from sparse original recommendation data. We select diffusion-based recommender systems CF-Diff and DiffRec for comparison with our SimDiff framework on the Tiktok, Amazon-Baby 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:

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

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.

It is evident that the percentage of changes for DiffRec and CF-Diff 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 iterative updates of the dynamic target. Moreover, the overall performance results in section 5.2 further confirm that SimDiff achieves significantly superior generative performance compared to the other two models. This observation reveals that static recovery targets constructed from sparse interaction data significantly limit the model's continuous learning capabilities, as they contain restricted information. Conversely, our proposed dynamic targets that can iteratively enrich their representations demonstrate superior performance in improving generation quality.

3 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 user and item. 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 $\hat{\boldsymbol{y}}_u = f(\mathcal{G}^{\mathcal{A}})$.

4 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 its semantic space, treating it as semantically rich noise. After that, we build a dynamic target to guide the model to denoise the latent variables and generate item representations that capture users' authentic interaction preferences. To iteratively update the targets, we develop a collaborative training strategy that optimize 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.

4.1 Signal Alignment Process

The auxiliary signal, which carry 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 pretrained 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:

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

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.

4.2 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.

4.2.1 *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 synthesizes 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 $\hat{\mathbf{G}}$ through a weighted fusion operation to obtain the latent variable $\mathbf{X}_T$ as follows:

$$\mathbf{X}_T = \mathbf{E}^i * \alpha + \hat{\mathbf{G}} * \beta, \tag{3}$$

where α and β denote 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.

4.2.2 *Dynamic Target Denoising Process.*

Although auxiliary signals in recommender systems contain rich semantic information, not all of them directly reflects 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

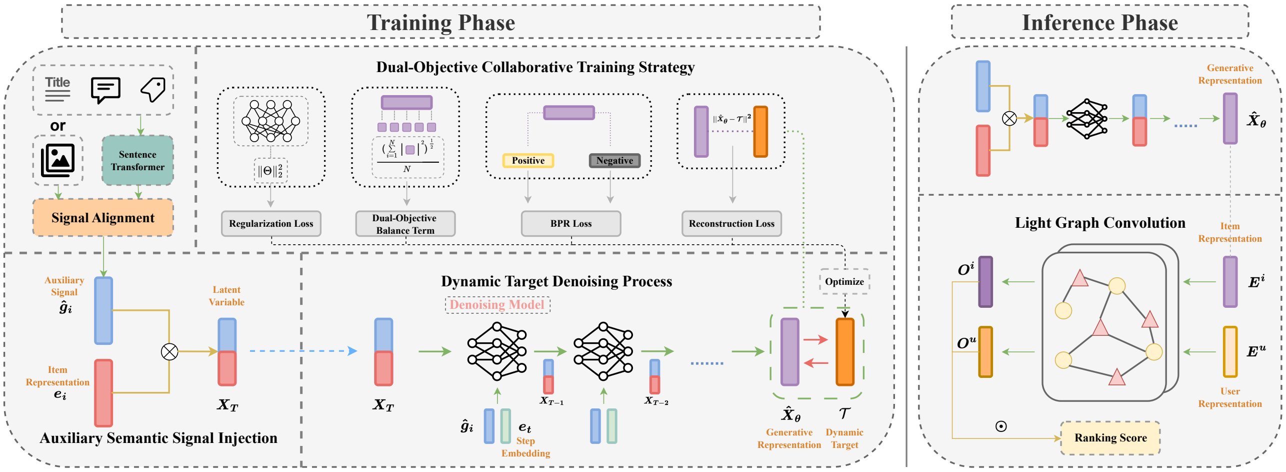

Training Phase
Inference Phase
Title
or
Sentence Transformer
Signal Alignment
Dual-Objective Collaborative Training Strategy
$\|\Theta\|_2^2$
$(\sum_{i=1}^{N} |\,|^2)^{\frac{1}{2}}$ / N
Positive
Negative
$\|\hat{x}_\theta - \mathcal{T}\|^2$
Regularization Loss
Dual-Objective Balance Term
BPR Loss
Reconstruction Loss
Optimize
Auxiliary Signal $\hat{g}_i$
Item Representation e_i
Latent Variable
X_T
Auxiliary Semantic Signal Injection
Dynamic Target Denoising Process
Denoising Model
X_T
X_{T-1}
X_{T-2}
$\hat{g}_i$
e_t Step Embedding
$\hat{X}_\theta$ Generative Representation
$\mathcal{T}$ Dynamic Target
Generative Representation
$\hat{X}_\theta$
Light Graph Convolution
Item Representation
O^i
E^i
O^u
E^u
User Representation
$\odot$
Ranking Score

Figure 3: The overall architecture of our proposed SimDiff, which involves injecting rich semantic information derived from text or image modalities into item representations. Dynamic targets are iteratively updated while guiding the generation of representations that contain authentic user preferences. The dual-objective collaborative training strategy continuously optimizes dynamic targets and other model parameters. 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.

to remove such noise while preserving the essential preference signals. As the existing recommender systems of using static user-item interactions and pre-trained representations as recovery targets which contain Insufficient user preference patterns constrains the model's generative ability, we introduce a dynamic learnable target as recovery target. This method enables continuous information enrichment throughout the model's training iterations, allowing for progressive accumulation of knowledge. By employing the previous epoch's training results as the recovery target for subsequent epochs, the target undergo adaptive updates, enabling more flexible representation learning and better capture of complex user preference.

In detail, we initialize the target representations $\mathcal{T}$, and employ an MLP structure as denoising model to process latent variables and generate item embeddings. Additionally, we preserve Sinusoidal Positional Encoding to help model learn the generation process of each step. The process is as follows:

$$\mathbf{X}_{T-1} = MLP(Contact[\mathbf{X}_T, \hat{\mathbf{G}}, \mathbf{E}_T^t]), \tag{4}$$

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, $\mathbf{E}_T^t \in \mathbb{R}^{N\times d_t}$ 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 $\mathbf{t}_i$ denote the dynamic target of item i, the t-th learning objective is:

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

Our objective is enabling the model to learn the progressive denoising process of latent variables, ultimately converging toward the recovery targets. Throughout this process, the dynamic targets continuously evolve, acquiring increasingly enriched representations. In the following subsection, we will provide a detailed explanation of the training strategy for updating this dynamic target.

4.2.3 *Dual-Objective Collaborative Training Strategy*.

The incorporation of auxiliary information enriches the generation process with semantic content. However, since the dynamic target is trained from initialization, it lacks co-occurrence relationships. To integrate user-item interaction patterns while training the generative model, we design a dual-objective collaborative training 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 formulation can be described as:

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

The loss term of reconstruction, denoted as $\mathcal{L}_r$, regulating 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 followed:

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

• **Dual-Objective Balance Term.** In the practical implementation, we observe an increasing divergence between reconstruction loss

Table 1: The comparison of analytical time complexity.

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

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 balance term $\mathcal{L}_c$ that specifically addresses the generation process. This supplementary loss term is motivated by the 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 constrain loss term $\mathcal{L}_c$ can be formally expressed through the following mathematical equation:

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

• **Total Optimization Objective.** 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:

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

Taking into account the previously outlined definitions, the consolidated optimization loss used in the training process for recommendation tasks is represented by:

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

here, Θ represents the learnable parameters of the model, with hyperparameters α_1 and α_2 controlling the relative strengths of the reconstruction and regularization terms.

4.3 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 followed:

$$\mathbf{O}^u = \bar{\mathcal{A}}_{u,*}\mathbf{H}^u,\ \mathbf{O}^i = \bar{\mathcal{A}}_{*,i}\mathbf{H}^i,\ \bar{\mathcal{A}}_{u,i} = \mathcal{A}_{u,i}/\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}, \tag{11}$$

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.

Table 2: Statistics of the datasets

Datasets	Office	Tiktok	Baby	Taobao	Electronics
#Users	4,905	9,308	19,445	12,539	32,886
#Items	2,420	6,710	7,050	8,735	52,974
#Int.	53,258	68,722	159,669	83,648	337,837
Sparsity	99.55%	99.88%	99.88%	99.92%	99.69%
TextDim	768	768	1024	–	300
ImageDim	4096	4096	4096	4096	4096

4.4 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 $O(BDd)$. 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 $O(kBHds)$. Given that our dual-objective collaborative training strategy simultaneously optimizes both the BPR loss and the reconstruction loss, their respective complexities are $O(2|\mathcal{E}|ds)$ and $O(Nds)$.
- **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 $O(TNHd)$. Moreover, since the normalized adjacency matrix has already been generated during the

Table 3: 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		Baby		Office		Taobao		Electronics	
	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
	@20	@20	@20	@20	@20	@20	@20	@20	@20	@20
	@50	@50	@50	@50	@50	@50	@50	@50	@50	@50
MF	0.0557	0.0235	0.0451	0.0185	0.0598	0.0232	0.0556	0.0207	0.0401	0.0155
	0.1046	0.0332	0.0899	0.0272	0.1178	0.0346	0.0983	0.0290	0.0620	0.0198
ENMF	0.1031	0.0395	0.0602	0.0287	0.1004	0.0500	0.1307	0.0630	0.0299	0.0139
	0.1656	0.0527	0.1055	0.0377	0.1729	0.0651	0.1813	0.0731	0.0512	0.0183
NGCF	0.0628	0.0245	0.0532	0.0226	0.0928	0.0400	0.1223	0.0523	0.0368	0.0163
	0.1166	0.0350	0.1002	0.0320	0.1684	0.0563	0.1902	0.0658	0.0593	0.0209
LightGCN	0.0907	0.0379	0.0715	0.0298	0.1215	0.0558	0.1502	0.0681	0.0394	0.0178
	0.1471	0.0491	0.1255	0.0409	0.2064	0.0702	0.2250	0.0830	0.0645	0.0229
SGL	0.0798	0.0342	0.0656	0.0297	0.1151	0.0549	0.1555	0.0748	0.0359	0.0175
	0.1308	0.0442	0.1090	0.0384	0.1838	0.0697	0.2107	0.0859	0.0561	0.0217
NCL	0.0898	0.0402	0.0742	0.0321	0.0966	0.0463	0.1558	0.0717	0.0435	0.0199
	0.1447	0.0510	0.1305	0.0433	0.1595	0.0594	0.2372	0.0880	0.0679	0.0249
LightGCL	0.0911	0.0435	0.0618	0.0231	0.1180	0.0531	0.1463	0.0649	0.0379	0.0163
	0.1190	0.0455	0.1158	0.0293	0.1942	0.0696	0.1986	0.0752	0.0528	0.0208
SCCF	0.0506	0.0216	0.0728	0.0349	0.1221	0.0520	0.1062	0.0540	0.0215	0.0103
	0.0883	0.0291	0.1136	0.0431	0.1963	0.0644	0.1388	0.0605	0.0332	0.0127
DiffRec	0.1036	0.0446	0.0713	0.0327	0.1159	0.0511	0.1492	0.0715	0.0236	0.0123
	0.1459	0.0536	0.1181	0.0422	0.1867	0.0704	0.2013	0.0824	0.0451	0.0189
DDRM-LightGCN	0.0145	0.0057	0.0118	0.0051	0.0133	0.0058	0.0139	0.0057	0.0033	0.0020
	0.0218	0.0072	0.0178	0.0063	0.0277	0.0088	0.0228	0.0075	0.0044	0.0022
DDRM-SGL	0.0281	0.0105	0.0151	0.0064	0.0381	0.0156	0.0821	0.0380	0.0060	0.0024
	0.0466	0.0147	0.0259	0.0086	0.0761	0.0237	0.1086	0.0433	0.0078	0.0028
CF-Diff	0.0665	0.0312	0.0751	0.0348	0.1028	0.0500	0.0529	0.0234	0.0099	0.0048
	0.1112	0.0402	0.1245	0.0449	0.1755	0.0658	0.0731	0.0274	0.0192	0.0067
GiffCF	0.1185	0.0462	0.0725	0.0323	0.1252	0.0537	0.1524	0.0659	0.0343	0.0138
	0.1687	0.0572	0.1253	0.0449	0.2084	0.0719	0.2084	0.0786	0.0509	0.0181
SimDiff	**0.1348***	**0.0588***	**0.0885***	**0.0389***	**0.1361***	**0.0606***	**0.1893***	**0.0783***	**0.0498***	**0.0217***
	0.1885*	**0.0694***	**0.1485***	**0.0507***	**0.2398***	**0.0808***	**0.2803***	**0.0965***	**0.0763***	**0.0278***
Improv.	13.77%	27.33%	17.84%	11.46%	8.75%	8.60%	21.50%	4.68%	14.48%	9.05%
	11.73%	21.37%	13.79%	12.98%	15.04%	12.38%	18.17%	9.66%	12.37%	11.65%

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 1. 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.

5 Experiments

5.1 Experimental Settings

5.1.1 *Datasets*.

We conduct experimental evaluations on five widely-used public recommendation datasets: TikTok, Amazon-Baby, Amazon-Office, Amazon-Electronics, and Taobao. The details of each dataset are shown in Table 2.

5.1.2 *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.

5.1.3 *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**) [14] and the efficient neural matrix factorization model ENMF [2]; (2) popular GNN-based models including **NGCF** [8] and **LightGCN** [7]; (3) recently proposed contrastive learning-based models that achieve high accuracy, specifically **SGL** [34], **NCL** [17], **SCCF** [35],

Table 4: Ablation Analysis Results

Dataset	Metric	(G, I+G)	(G, I)	(I, I+G)	Pretrain	SimDiff
TikTok	Recall@20	0.1219	0.1204	0.1206	0.0899	**0.1348**
	Recall@50	0.1909	0.1903	0.1909	0.1449	**0.1885**
	NDCG@20	0.0520	0.0513	0.0511	0.0361	**0.0588**
	NDCG@50	0.0660	0.0655	0.0653	0.0469	**0.0694**
Baby	Recall@20	0.0698	0.0700	0.0699	0.0482	**0.0885**
	Recall@50	0.1221	0.1229	0.1246	0.0915	**0.1485**
	NDCG@20	0.0278	0.0288	0.0282	0.0204	**0.0389**
	NDCG@50	0.0384	0.0395	0.0393	0.0291	**0.0507**
Office	Recall@20	0.1314	0.1318	0.1349	0.1275	**0.1361**
	Recall@50	0.2240	0.2146	0.2301	0.2166	**0.2398**
	NDCG@20	0.0549	0.0518	0.0553	0.0587	**0.0606**
	NDCG@50	0.0741	0.0691	0.0750	0.0771	**0.0808**
Taobao	Recall@20	0.1456	0.1453	0.1669	0.1775	**0.1893**
	Recall@50	0.2341	0.2415	0.2519	0.2581	**0.2803**
	NDCG@20	0.0556	0.0569	0.0681	0.0789	**0.0783**
	NDCG@50	0.0732	0.0760	0.0850	0.0949	**0.0965**
Electronics	Recall@20	0.0412	0.0426	0.0410	0.0424	**0.0498**
	Recall@50	0.0682	0.0691	0.0688	0.0690	**0.0763**
	NDCG@20	0.0183	0.0191	0.0184	0.0185	**0.0217**
	NDCG@50	0.0239	0.0245	0.0241	0.0240	**0.0278**

and **LightGCL** [1]; and (4) state-of-the-art diffusion-based generative models from the past two years, namely **DiffRec** [29], **DDRM** [40], **GiffCF** [46], and **CF-Diff** [10].

5.1.4 *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 dropout rate, crucial for preventing overfitting, is uniformly sampled between 0.0 and 0.5. The reconstructon alpha 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.

5.2 Performance Comparison

Table 3 presents a comparative analysis of our proposed model against various baseline models across five 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 oversmoothing, 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.

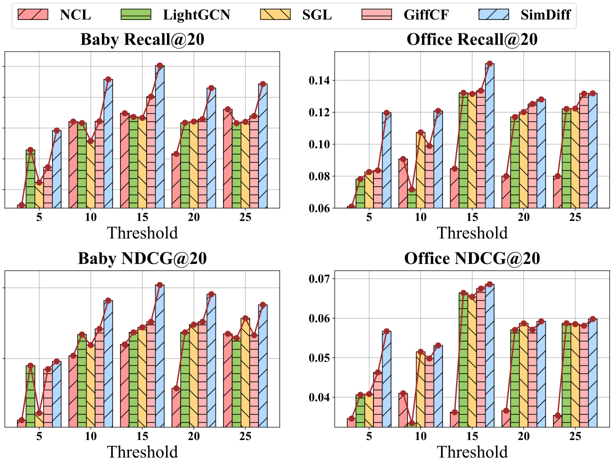


Figure 4: Performance comparison over Amazon-Baby and Amazon-Office between SimDiff and other outstanding baseline models in cold-start recommendation scenario

- 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 dynamic learnable targets, trained through a dual-objective collaborative strategy with self-iterative updates, overcome the limitation of static targets, significantly improving generation performance.

5.3 Ablation Analysis

Table 4 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 and item representation is utilized as the training target for the generative process. The variants **(G, I)** and **(G, I+G)** follow similar patterns. **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.

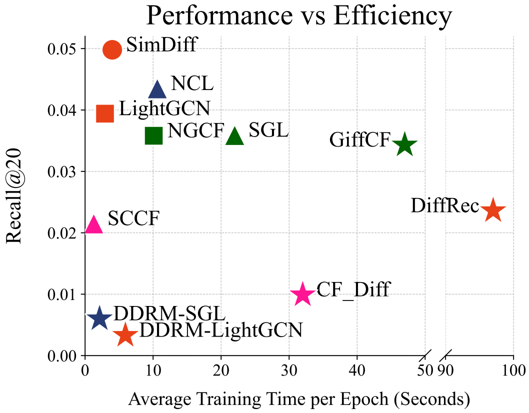


Figure 5: Performance versus Efficiency Analysis on Amazon-Office, Amazon-Baby, Amazon-Electronics and Taobao datasets. Performance strength and training efficiency increase towards the upper left direction.

Notably, when the generative target is set to invariable pretrained representations, we observe a significant performance degradation. This finding highlights the substantial utility of learnable training targets in our framework. The empirical evidence strongly supports the advantages of our approach in leveraging dynamic, learnable targets over static, pretrained representations.

5.4 Indepth Model Analysis

5.4.1 *Cold-start Recommendation*.

As mentioned in introduction, data sparsity in recommendation is a critical promblem. To prove out proposed SimDiff has the advantage to solve this issue, we conduct cold-start experiments on Amazon-Baby and Amazon-Office dataset, wherein most users has scarce interactions with items. Figure 4 shows the results of cold-start recommendation.

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 baby 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 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.

5.4.2 *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 were evaluated using the same GPU with single-process execution. As illustrated in Figure 5, 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, shows 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 5: Auxiliary Signal Analysis

Signal	Metric	TikTok	Baby	Office	Electronics
G = Image	Recall@20	0.1310	0.0833	0.1327	0.0469
	Recall@50	0.1933	0.1371	0.2246	0.0768
	NDCG@20	0.0588	0.0364	0.0540	0.0209
	NDCG@50	0.0713	0.0474	0.0731	0.0271
G = Text	Recall@20	0.1348	0.0885	0.1361	0.0498
	Recall@50	0.1885	0.1485	0.2398	0.0763
	NDCG@20	0.0588	0.0389	0.0606	0.0217
	NDCG@50	0.0694	0.0507	0.0808	0.0278

5.4.3 *Auxiliary Signal Analysis*.

In our results of section 5.2, 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 four 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 showed in Table 5.

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.

5.5 Analysis of Dual-Objective Balance Term

In this subsection, we demonstrate the core idea of adding the dual-objective balance term. We first present a comparison of the results before and after incorporating this term in Table 6. It is evident that the addition of this term 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 4.2.3, we implement a dual-objective collaborative training strategy to simultaneously optimize both the generation process and recommendation task objectives. The curves

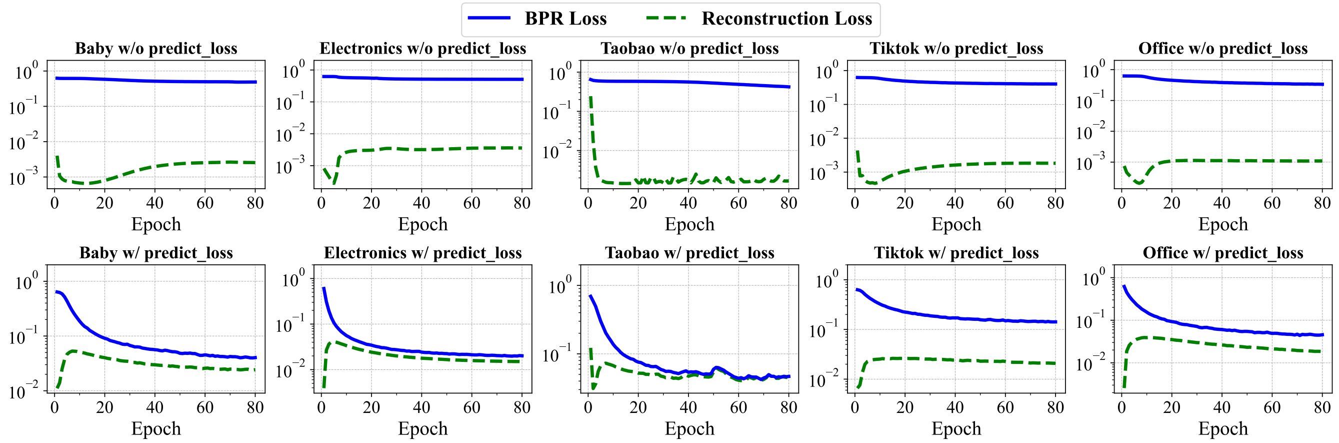


Figure 6: Comparison of w/o CBT and SimDiff

Table 6: Effectiveness of Dual-Objective Balance Term

Dataset	@K	w/o CBT		SimDiff	
		Recall	NDCG	Recall	NDCG
TikTok	**@20**	0.0899	0.0361	**0.1348**	**0.0588**
	@50	0.1449	0.0469	**0.1885**	**0.0694**
Baby	**@20**	0.0482	0.0204	**0.0885**	**0.0389**
	@50	0.0915	0.0291	**0.1485**	**0.0507**
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**

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 dual-objective balance term. Before its implementation, the two loss components differed greatly in magnitude. With this term integrated, the losses balanced to similar values, allowing stable training of the denoising model. This balanced optimization approach significantly enhances both the generative capabilities and overall model performance, as evidenced by our experimental results.

6 Related Work

6.1 Collaborative Filtering

Research in recommender systems evolved from content-based and collaborative filtering approaches in the 1990s to matrix factorization (MF) techniques [13, 14, 25] during the Netflix Prize era. While MF methods captured preference patterns via latent factors, their limitations with data sparsity and non-linear relationships led to Neural Collaborative Filtering (NCF) [8] and subsequent Graph Neural Network approaches, notably NGCF [30] for message passing and LightGCN [7], which simplified graph convolution operations for better efficiency. The field then witnessed significant advancement through contrastive learning methods, inspired by SimCLR [4], with Self-supervised Graph Learning (SGL) [34] introducing graph augmentation techniques and Neighbor Contrastive Learning (NCL) [17] refining negative sampling. Latest developments include SCCF [35] unifying graph convolution with contrastive learning, RGCL [28] employing adversarial perturbations, and RecDCL [43] implementing a dual framework for batch-wise and feature-wise contrastive objectives. This evolution underscores the growing emphasis on learning robust and discriminative representations while maintaining computational efficiency in recommender systems.

6.2 Diffusion Based Recommendation

Diffusion models have gained success since DDPM [9], which learns a denoising process through adding and removing Gaussian noise. Extensions to improve sampling efficiency include non-Markovian processes [26] and conditional generation [5]. In recommender systems, DiffRec [29] applied diffusion to user-item interaction graphs, while DreamRec [40] incorporated user history. DDRM [44] introduced mutual conditioning between users and items during the diffusion process, and GiffCF [46] simulated the heat equation on graphs. These works highlight the potential of diffusion models in modeling complex user-item interactions.

7 Conclusion

In this work, we propose a novel diffusion framework called SimDiff for recommender system. We replace the randomly sampled Gaussian noise addition by injecting auxiliary signal derived from modal feature to representations, which introduce rich semantic information to sparse data. In order to improve the generative effect, we build a dynamic target and update iteratively by collaboratively training the denoising model and optimizing BPR loss. 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] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *ICLR*.

[2] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient neural matrix factorization without sampling for recommendation. *TOIS* 38, 2 (2020), 1–28.

[3] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yiqun Liu, Yixing Fan, and Xueqi Cheng. 2023. A unified generative retriever for knowledge-intensive language tasks via prompt learning. In *SIGIR*. ACM, 1448–1457.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. PMLR, 1597–1607.

[5] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion Models Beat GANs on Image Synthesis. In *NeurIPS*, Vol. 34. Curran Associates, Inc., 8780–8794.

[6] Shen Gao, Jiabao Fang, Quan Tu, Zhitao Yao, Zhumin Chen, Pengjie Ren, and Zhaochun Ren. 2024. Generative News Recommendation. In *Proceedings of the ACM on Web Conference 2024*. ACM, 3444–3453.

[7] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. ACM, 639–648.

[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. ACM, 173–182.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *NeurIPS* 33 (2020), 6840–6851.

[10] Yu Hou, Jin-Duk Park, and Won-Yong Shin. 2024. Collaborative filtering based on diffusion models: Unveiling the potential of high-order connectivity. In *SIGIR*. ACM, 1360–1369.

[11] Cong Jiang, Zhongde Chen, Bo Zhang, Yankun Ren, Xin Dong, Lei Cheng, Xinxing Yang, Longfei Li, Jun Zhou, and Linjian Mo. 2024. GATS: Generative Audience Targeting System for Online Advertising. In *SIGIR*. ACM, 2920–2924.

[12] Yangqin Jiang, Yuhao Yang, Lianghao Xia, and Chao Huang. 2024. Diffkg: Knowledge graph diffusion model for recommendation. In *WSDM*. ACM, 313–321.

[13] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*. Association for Computing Machinery, 426–434.

[14] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* (2009), 30–37.

[15] Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Ming-Yu Liu, Kai Li, and Song Han. 2024. DistriFusion: Distributed Parallel Inference for High-Resolution Diffusion Models. In *CVPR*. 7183–7193.

[16] Dengtian Lin, Liqiang Jing, Xuemeng Song, Meng Liu, Teng Sun, and Liqiang Nie. 2023. Adapting generative pretrained language model for open-domain multimodal sentence summarization. In *SIGIR*. ACM, 195–204.

[17] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM web conference 2022*. ACM, 2320–2329.

[18] Xiangyue Liu, Han Xue, Kunming Luo, Ping Tan, and Li Yi. 2024. GenN2N: Generative NeRF2NeRF Translation. In *CVPR*. 5105–5114.

[19] Zhiding Liu, Jiqian Yang, Mingyue Cheng, Yucong Luo, and Zhi Li. 2024. Generative pretrained hierarchical transformer for time series forecasting. In *SIGKDD*. ACM, 2003–2013.

[20] Jing Long, Guanhua Ye, Tong Chen, Yang Wang, Meng Wang, and Hongzhi Yin. 2024. Diffusion-based cloud-edge-device collaborative learning for next POI recommendations. In *SIGKDD*. ACM, 2026–2036.

[21] Xiao Long, Liansheng Zhuang, Aodi Li, Houqiang Li, and Shafei Wang. 2024. Fact Embedding through Diffusion Model for Knowledge Graph Completion. In *Proceedings of the ACM on Web Conference 2024*. ACM, 2020–2029.

[22] Haokai Ma, Ruobing Xie, Lei Meng, Yimeng Yang, Xingwu Sun, and Zhanhui Kang. 2024. Seedrec: sememe-based diffusion for sequential recommendation. In *Proceedings of IJCAI*. 1–9.

[23] Haokai Ma, Yimeng Yang, Lei Meng, Ruobing Xie, and Xiangxu Meng. 2024. Multimodal Conditioned Diffusion Model for Recommendation. In *Companion Proceedings of the ACM on Web Conference 2024*. ACM, 1733–1740.

[24] Trung-Kien Nguyen and Yuan Fang. 2024. Diffusion-based Negative Sampling on Graphs for Link Prediction. In *Proceedings of the ACM on Web Conference 2024*. ACM, 948–958.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. AUAI Press.

[26] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *ICML*.

[27] Li Sun, Jingbin Hu, Suyang Zhou, Zhenhao Huang, Junda Ye, Hao Peng, Zhengtao Yu, and Philip Yu. 2024. Riccinet: Deep clustering via a riemannian generative model. In *Proceedings of the ACM on Web Conference 2024*. ACM, 4071–4082.

[28] Jiakai Tang, Sunhao Dai, Zexu Sun, Xu Chen, Jun Xu, Wenhui Yu, Lantao Hu, Peng Jiang, and Han Li. 2024. Towards Robust Recommendation via Decision Boundary-aware Graph Contrastive Learning. In *SIGKDD*. ACM, 2854–2865.

[29] Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. 2023. Diffusion recommender model. In *SIGIR*. ACM, 832–841.

[30] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*. ACM, 165–174.

[31] Yuhao Wang, Ziru Liu, Yichao Wang, Xiangyu Zhao, Bo Chen, Huifeng Guo, and Ruiming Tang. 2024. Diff-MSR: A Diffusion Model Enhanced Paradigm for Cold-Start Multi-Scenario Recommendation. In *WSDM*. ACM, 779–787.

[32] Ye Wang, Jiahao Xun, Minjie Hong, Jieming Zhu, Tao Jin, Wang Lin, Haoyuan Li, Linjun Li, Yan Xia, Zhou Zhao, et al. 2024. EAGER: Two-Stream Generative Recommender with Behavior-Semantic Collaboration. In *SIGKDD*. ACM, 3245–3254.

[33] Xumeng Wen, Han Zhang, Shun Zheng, Wei Xu, and Jiang Bian. 2024. From supervised to generative: A novel paradigm for tabular deep learning with large language models. In *SIGKDD*. ACM, 3323–3333.

[34] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. ACM, 726–735.

[35] Yihong Wu, Le Zhang, Fengran Mo, Tianyu Zhu, Weizhi Ma, and Jian-Yun Nie. 2024. Unifying Graph Convolution and Contrastive Learning in Collaborative Filtering. In *SIGKDD*. ACM, 3425–3436.

[36] Yiyan Xu, Wenjie Wang, Fuli Feng, Yunshan Ma, Jizhi Zhang, and Xiangnan He. 2024. Diffusion models for generative outfit recommendation. In *SIGIR*. ACM, 1350–1359.

[37] Shuchen Xue, Zhaoqiang Liu, Fei Chen, Shifeng Zhang, Tianyang Hu, Enze Xie, and Zhenguo Li. 2024. Accelerating Diffusion Sampling with Optimized Time Steps. In *CVPR*. 8292–8301.

[38] Ling Yang, Haotian Qian, Zhilong Zhang, Jingwei Liu, and Bin Cui. 2024. Structure-Guided Adversarial Training of Diffusion Models. In *CVPR*. 7256–7266.

[39] Yonghui Yang, Zhengwei Wu, Le Wu, Kun Zhang, Richang Hong, Zhiqiang Zhang, Jun Zhou, and Meng Wang. 2023. Generative-contrastive graph learning for recommendation. In *SIGIR*. ACM, 1117–1126.

[40] Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. 2024. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. *NeurIPS* 36 (2024).

[41] Zixuan Yi, Xi Wang, and Iadh Ounis. 2024. A Directional Diffusion Graph Transformer for Recommendation. In *SIGIR*. ACM.

[42] Hao Zeng, Jiaqi Wang, Avirup Das, Jun He, Kunpeng Han, Haoyuan Hu, and Mingfei Sun. 2024. Effective Generation of Feasible Solutions for Integer Programming via Guided Diffusion. In *SIGKDD*. ACM, 4107–4118.

[43] Dan Zhang, Yangliao Geng, Wenwen Gong, Zhongang Qi, Zhiyu Chen, Xing Tang, Ying Shan, Yuxiao Dong, and Jie Tang. 2024. RecDCL: Dual Contrastive Learning for Recommendation. In *Proceedings of the ACM on Web Conference 2024*. ACM, 3655–3666.

[44] Jujia Zhao, Wang Wenjie, Yiyan Xu, Teng Sun, Fuli Feng, and Tat-Seng Chua. 2024. Denoising diffusion recommender model. In *SIGIR*. ACM, 1370–1379.

[45] Ting Zhong, Jienan Zhang, Zhangtao Cheng, Fan Zhou, and Xueqin Chen. 2024. Information Diffusion Prediction via Cascade-Retrieved In-context Learning. In *SIGIR*. ACM, 2472–2476.

[46] Yunqin Zhu, Chao Wang, Qi Zhang, and Hui Xiong. 2024. Graph signal diffusion model for collaborative filtering. In *SIGIR*. ACM, 1380–1390.