Diffusion-augmented Graph Contrastive Learning for Collaborative Filtering

Fan Huang
School of Data Science and Engineering
East China Normal University
fanhuang@stu.ecnu.edu.cn

Jianxiang Yu
School of Data Science and Engineering
East China Normal University
jianxiangyu@stu.ecnu.edu.cn

Wei Wang*
School of Data Science and Engineering
East China Normal University
wwang@dase.ecnu.edu.cn

Abstract

Graph-based collaborative filtering has been established as a prominent approach in recommendation systems, leveraging the inherent graph topology of user-item interactions to model high-order connectivity patterns and enhance recommendation performance. Recent advances in Graph Contrastive Learning (GCL) have demonstrated promising potential to alleviate data sparsity issues by improving representation learning through contrastive view generation and mutual information maximization. However, existing approaches lack effective data augmentation strategies. Structural augmentation risks distorting fundamental graph topology, while feature-level perturbation techniques predominantly employ uniform noise scales that fail to account for node-specific characteristics. To solve these challenges, we propose Diffusion-augmented Contrastive Learning (DGCL), an innovative framework that integrates diffusion models with contrastive learning for enhanced collaborative filtering. Our approach employs a diffusion process that learns node-specific Gaussian distributions of representations, thereby generating semantically consistent yet diversified contrastive views through reverse diffusion sampling. DGCL facilitates adaptive data augmentation based on reconstructed representations, considering both semantic coherence and node-specific features. In addition, it explores unrepresented regions of the latent sparse feature space, thereby enriching the diversity of contrastive views. Extensive experimental results demonstrate the effectiveness of DGCL on three public datasets. Code is available at https://github.com/huangfan0/DGCL.

1 Introduction

Collaborative Filtering (CF) is fundamental to recommendation systems, predicting user preferences from historical user-item interactions Su and Khoshgoftaar [2009], Koren et al. [2021]. Traditional methods primarily used node embedding techniques such as matrix factorization (MF) Koren et al. [2009] or graph-based approaches like DeepWalk Perozzi et al. [2014]. More recently, graph neural networks (GNNs), especially graph convolutional networks (GCN) Kipf and Welling [2016], have transformed the field through recursive neighborhood aggregation. GNN-based CF models including

*Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: New Perspectives in Graph Machine Learning.

NGCF Wang et al. [2019] and LightGCN He et al. [2020] explicitly encode multi-hop connectivity via message passing, effectively capturing higher-order collaborative signals beyond direct interactions.

Despite these advances, graph-based collaborative filtering faces data sparsity, leading to suboptimal representations with poor generalization beyond observed interactions. To address this, GCL Jaiswal et al. [2020], Yu et al. [2023], Jiang et al. [2023] has emerged as a promising self-supervised paradigm that mitigates popularity bias and improves generalization. For example, SimGCL Yu et al. [2022] perturbs node embeddings with uniform noise to regularize representation uniformity and reduce bias. Existing GCL methods mainly adopt two augmentation strategies: structural augmentation (e.g., node/edge dropout, subgraph sampling) and feature augmentation (e.g., feature masking, noise injection, or clustering).

However, conventional augmentations face two key limitations. Structural perturbations risk disrupting critical topological dependencies (e.g., key nodes or user-item interactions), while feature-level methods like uniform noise addition ignore nodes' heterogeneous semantics and unique features Yang et al. [2023] by applying identical distortion to both popular and long-tail items. This generates semantically inconsistent views that impair representation learning, making it a challenge to design augmentations that preserve graph semantics while diversifying views.

To address this, we propose a Diffusion-augmented Graph Contrastive Learning (DGCL) framework that leverages diffusion models to generate adaptive augmentations. Specifically, DGCL models node-specific Gaussian distributions in the embedding space, allowing adaptive view sampling conditioned on each node's semantic context. A forward diffusion process adds controlled noise, while reverse denoising produces diverse yet semantically consistent contrastive views. This node-adaptive mechanism synthesizes high-quality augmentations that preserve semantic coherence. Moreover, the diffusion process uncovers latent information in sparse interaction data, enabling exploration of under-represented regions without altering graph topology. This enhances both the diversity and quality of contrastive pairs. Thus, DGCL maintains structural integrity while providing granular, node-specific augmentations—overcoming key limitations of conventional GCL. Experiments show that DGCL effectively balances robustness and semantic coherence, advancing the state-of-the-art in graph-based collaborative filtering. The main contributions of this work are summarized as follows:

- We propose a novel Diffusion-augmented Graph Contrastive Learning (DGCL) framework, which incorporates the diffusion model into contrastive learning for collaborative filtering.
- We design a data augmentation scheme, in which the diffusion model was used to generate high-quality contrastive views that consider semantic correlation and node-specific features.
- Experiments on three public datasets confirm the superior effectiveness of the DGCL.

2 Related Work

2.1 Graph Recommendation

Contrastive learning is a key self-supervised learning (SSL) Yu et al. [2023] method for addressing data sparsity and popularity bias in recommendation systems Lin et al. [2022] Wang et al. [2022]. LightGCN He et al. [2020] simplified graph convolution by removing non-linear transformations and focusing on high-order neighborhood aggregation. SGL Wu et al. [2021] enhanced robustness through structural augmentations such as stochastic node/edge dropout, contrasting multiple subgraph views via a shared encoder. SimGCL Yu et al. [2022] further improved bias reduction and representation uniformity by perturbing embeddings with uniform noise. Despite exhibiting the versatility of GCL, these methods still face challenges in data augmentation and constructing effective contrastive views.

2.2 Diffusion-based Graph Learning

The diffusion model has emerged as a leading approach in image generation, spurred by advances in denoising diffusion probabilistic models (DDPM) Ho et al. [2020] and Score-based Generative Models (SGM) Song et al. [2020]. Recently, these models have inspired innovative frameworks for recommendation systems. Early efforts, such as DiffRec Wang et al. [2023], reformulated user-item interactions as a denoising process to preserve personalized signals through noise injection and reconstruction. Subsequent extensions enhanced its scalability. CF-Diff Hou et al. [2024] improved upon DiffRec by explicitly modeling high-order graph connectivity via multi-step denoising to

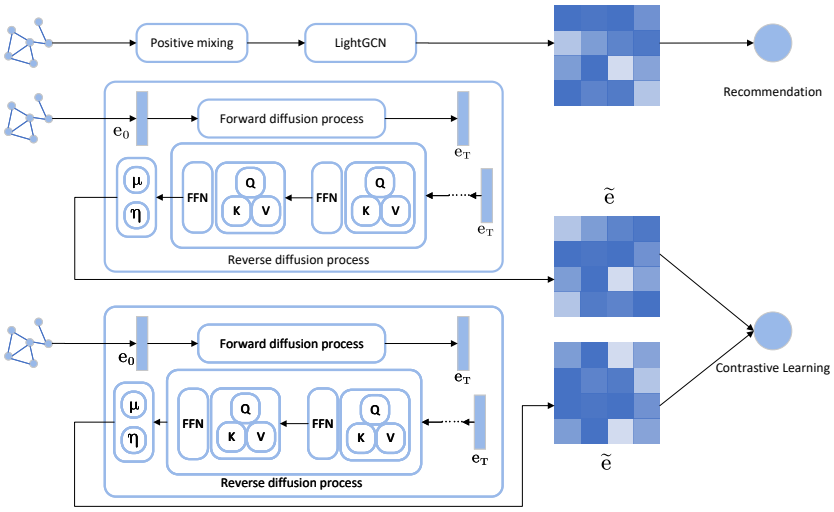


Figure 1: Overall framework of DGCL.

capture collaborative patterns. BSPM Choi et al. [2023] introduced an image-inspired blurring-and-sharpening paradigm to extract noise-robust signals, DiffKG Jiang et al. [2024] integrated knowledge graphs to leverage semantic information, and GiffCF Zhu et al. [2024] defined a diffusion process over item-item graphs to model interaction distributions. SCONE Lee et al. [2025] employed SGM in recommender systems to generate dynamic contrastive views and diverse hard negative samples through stochastic sampling processes. Different from the above diffusion-based graph learning, our DGCL model focuses on data augmentation and generates semantically consistent yet diverse contrastive views in collaborative filtering.

3 Methods

As illustrated in Figure 1, the proposed DGCL framework comprises three modules: graph collaborative learning, diffusion-based augmentation, and contrastive learning. The graph collaborative learning module employs a LightGCN encoder to capture high-order neighbor information and learn latent embeddings, incorporating a novel negative sampling strategy to improve discriminative power. The diffusion augmentation module applies a dual diffusion process: Gaussian noise is systematically injected via forward diffusion, and contrastive views are generated by sampling from the learned posterior distributions. Finally, the contrastive learning module facilitates cross-view comparison between users and items through a dual-channel architecture. The entire framework is optimized with a joint loss, combining collaborative filtering and contrastive objectives.

3.1 Graph Collaborative Relation Learning

Given initial user and item embeddings e_u and e_i, LightGCN learns node representations through iterative message propagation and aggregation across layers. To improve negative sample quality, we incorporate positive mixing Huang et al. [2021], which generates challenging synthetic negatives by blending positive and negative embeddings:

$$\mathbf{e}_{i^-}^{\prime(l)} = \alpha^{(l)}\mathbf{e}_{i^+}^{(l)} + \left(1-\alpha^{(l)}\right)\mathbf{e}_{i^-}^{(l)}, \alpha^{(l)} \in (0,1), \tag{1}$$

where $\mathbf{e}_{i^+}^{(l)}$ and $\mathbf{e}_{i^-}^{(l)}$ denote the positive and negative item embeddings at layer l, respectively. This produces harder negatives closer to positives in the embedding space, improving the model's discriminative power and ranking accuracy. Independent layer-wise mixing introduces multi-level semantic interference, further enhancing robustness.

3.2 Diffusion Augmentation Module

Unlike conventional diffusion methods that perturb adjacency matrices, our approach applies diffusion in the latent embedding space to produce augmented contrastive views. The module comprises two phases: a forward noise injection process and a parameterized reverse generation process.

Forward Diffusion Process. Given nodes representation e, including the user representation e_u and item representation e_i, we inject Gaussian noise over T steps according to a variance schedule $\beta_t{}_{t=1}^{T}$. The forward process is a Markov chain defined as:

$$q\left(e_{1:T} \mid e_0\right) = \prod_{t=1}^{T} q\left(e_t \mid e_{t-1}\right), \tag{2}$$

where e_0 represents the original nodes' embedding and e_T denotes the noise-perturbed representation. The specific condition distribution for each step is:

$$q\left(e_t \mid e_{t-1}\right) = \mathcal{N}\left(e_t; \sqrt{1-\beta_t}e_{t-1}, \beta_t \mathbf{I}\right), \tag{3}$$

where β controls the Gaussian noise scales at each time step t, $\mathcal{N}$ refers to the Gaussian distribution. By exploiting the reparameterization trick Kingma et al. [2013], the noised representation can be expressed as $\boldsymbol{e}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{e}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t$, where $\alpha_t = 1-\beta_t, \bar{\alpha}_t = \prod_{t'=1}^{t}\alpha_{t'},$ and $\epsilon \sim \mathcal{N}(0, I)$. For large T, e_T approaches standard Gaussian noise. The noisy embeddings are used to train the model and estimate Gaussian parameters.

Reverse Diffusion Process. The reverse process generates data through iterative denoising, modeled as a parameterized Markov chain where each step follows a Gaussian distribution:

$$p_\theta(e_{t-1}|e_t) = \mathcal{N}\left(e_{t-1}; \mu_\theta(e_t, t), \sigma_t^2\mathbf{I}\right), \tag{4}$$

where μ_θ, σ_t are the learned parameters.We use a two-layer Transformer Vaswani et al. [2017] encoder with multi-head attention and feed-forward networks. To incorporate temporal information, diffusion time steps are embedded using sinusoidal positional encoding:

$$PE(t, 2i) = \sin(\frac{t}{10000^{2i/d}}), \quad PE(t, 2i+1) = \cos(\frac{t}{10000^{2i/d}}). \tag{5}$$

Time embeddings are integrated into the feature space via feature-wise linear modulation Perez et al. [2018], which can be formulated as:

$$\gamma, \eta = \text{TimeMLP}(t), h = (\gamma + 1)e + \eta \tag{6}$$

Denoised user/item representations are then obtained through self-attention and feed-forward layers:

$$h^{(l+1)} = \text{LayerNorm}(h^{(l)} + \text{Multi\_Attention}(h^{(l)})) \tag{7}$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{8}$$

where $Q = W_Q h, K = W_K h, V = W_V h$. This allows the model to predict Gaussian noise parameters that reconstruct node representations throughout the diffusion process.

Diffusion Loss. The diffusion augmentation module learns the parameters of the Gaussian distribution of latent user and item embeddings e_0. Instead of directly predicting noise, DGCL recovers the original embedding structure. The objective of DGCL is:

$$\mathcal{L}_{diff}(\theta) = \mathbb{E}_{t,e_0}\left[\left\|e_0 - f_\theta(e_t, t)\right\|^2\right], \tag{9}$$

where f_θ is the mapping function that predicts augmented representations. Training pseudo-code is provided in appendix A Algorithm 1

Contrastive View Inference. The embedding encoder $f(x_t, t)$, derived from the diffusion process, generates augmented contrastive view embeddings. Our model estimates distribution parameters to iteratively approximate the posterior $p_\theta(e_{t-1}|e_t)$, from which refined representations are inferred. The posterior mean and covariance are computed as:

$$\mu_\theta(e_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(e_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(e_t, t)\right), \tag{10}$$

$$\sigma_t^2(t) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{11}$$

Therefore, the augmented contrastive user views $\widetilde{e}_u$ and item views $\widetilde{e}_i$ are iteratively generated via:

$$e_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(e_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(e_t, t) \right) + \sigma_t z, \tag{12}$$

This produces high-quality synthetic views that preserve semantic and personalized features while uncovering underrepresented regions of the embedding space, thereby mitigating sparsity and increasing view diversity. The full inference process is described in Appendix A Algorithm 2.

3.3 Contrastive Learning Module

In DGCL, we employ separate diffusion augmentation modules to produce diverse contrastive views for users and items. Each augmented embedding remains aligned with its original node's feature space. The user and item contrastive losses are:

$$\mathcal{L}_{cl}^{U} = \sum_{u \in \mathcal{B}_u} -\log \frac{\exp(\widetilde{\mathbf{e}}_u'^{\top} \widetilde{\mathbf{e}}_{\mathbf{u}}'' / \tau_1)}{\sum_{j \in \mathcal{B}_u} \exp(\widetilde{\mathbf{e}}_u'^{\top} \widetilde{\mathbf{e}}_{\mathbf{j}}'' / \tau_1)}, \mathcal{L}_{cl}^{I} = \sum_{i \in \mathcal{B}_i} -\log \frac{\exp(\widetilde{\mathbf{e}}_{\mathbf{i}}'^{T} \widetilde{\mathbf{e}}_{\mathbf{i}}'' / \tau_1)}{\sum_{j \in \mathcal{B}_i} \exp(\widetilde{\mathbf{e}}_{\mathbf{i}}'^{T} \widetilde{\mathbf{e}}_{\mathbf{j}}'' / \tau_1)}, \tag{13}$$

where τ_1 is the temperature hyperparameter, scaling the similarity scores between embeddings.

3.4 Model Training

DGCL involves two independent training objectives: one for learning the diffusion process to generate augmented contrastive views, optimized via Eq. 9, and another for jointly optimizing the recommendation and contrastive modules. The overall objective combines these through a joint loss:

$$\mathcal{L}_{joint} = \mathcal{L}_{rec} + \lambda \mathcal{L}_{cl}, \tag{14}$$

where the L_{rec} is the Bayesian Personalized Ranking (BPR) loss:

$$\mathcal{L}_{rec} = -\log \left(\sigma \left(\mathbf{e}_u^{\top} \mathbf{e}_i - \mathbf{e}_u^{\top} \mathbf{e}_j \right) \right), \tag{15}$$

and L_{cl} denotes the contrastive loss which includes the $\mathcal{L}_{cl}^{U}$ and $\mathcal{L}_{cl}^{I}$ as defined in the Eq. 13.

4 Theoretical Analysis

In this section, we provide an elaborate theoretical analysis on why diffusion provides semantic but diverse augmentation views. For detail proof, please refer to the Appendix B.

theorem 4.1 (Manifold Reconstruction)**.** *Suppose the reverse model satisfies $p_\theta(e_{t-1} \mid e_t) = q(e_{t-1} \mid e_t, e_0)$ for all t and that the prior used at $t = T$ equals $q(e_T)$. Then the marginal distribution of the reconstructed samples equals the data distribution:$p_\theta(\hat{e}_0) = p_{data}(e_0)$. Consequently, the support of $p_\theta(\hat{e}_0)$ coincides with the semantic manifold $\mathcal{M}$.*

theorem 4.2 (Semantic Consistency)**.** *Under the assumed forward process and with the reverse model matching the true posterior, the sampled augmented embedding $\hat{e}_0$ satisfies $\mathbb{E}[\hat{e_0} \mid e_0] = e_0$ i.e, the reverse-sampled view is mean-unbiased for the original embedding. Moreover, the conditional mean-squared error equals the posterior variance trace: $\mathbb{E}\big[\|\hat{e_0} - e_0\|^2 \mid e_0\big] = \mathrm{Tr}\big(\mathrm{Var}(e_0 \mid e_0)\big)$ The posterior variance can be bounded in terms of the diffusion schedule β_t; in particular there exists a constant $C(d)$ such that $\mathbb{E}\left[\|\hat{e}_0 - e_0\|^2\right] \leq C(d) \sum_{t=1}^{T} \beta_t$.*

proposition 4.3 (Controlled Diversity)**.** *Let $\hat{e}_0^{(1)}$ and $\hat{e}_0^{(2)}$ be two independent samples drawn from the reverse model conditioned on the same e_0. Then $\mathbb{E}\left[\|\hat{e}_0^{(1)} - \hat{e}_0^{(2)}\|^2 \mid e_0\right] = 2\, Tr\left(Var(\hat{e}_0 \mid e_0)\right)$. In particular, when the posterior variance is nonzero the reverse process produces distinct views in expectation, and the magnitude of diversity is controlled by the posterior covariance. Because DGCL estimates node-adaptive posterior covariances, diversity is node-adaptive.*

5 Experiments

In this section, we conduct extensive experiments to evaluate the performance of DGCL on three benchmark datasets and analyze the key module of the model.

5.1 Experimental Setup

Datasets. The experience employs three public datasets in different scenarios. (1) Douban-Book Yu et al. [2023]. (2) Gowalla[2] Wang et al. [2022]. (3) Amazon-Kindle Yu et al. [2023]. For detail experiment parameters and hyper-parameter sensitivity Analysis, please refer to Appendix C.

Table 1: DGCL Performance Comparison with different methods on three datasets.

Models	Douban-Book				Gowalla				Amazon-Kindle			
	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
BPR-MF Koren et al. [2009]	0.0869	0.0949	0.1296	0.1045	0.1158	0.0833	0.1695	0.0988	0.10873	0.0801	0.14949	0.0923
LighGCN He et al. [2020]	0.1042	0.1195	0.1516	0.1278	0.1262	0.0876	0.1776	0.1152	0.1425	0.1063	0.1906	0.1208
SGL Wu et al. [2021]	0.1103	0.1357	0.1551	0.1419	0.1255	0.1371	0.1783	0.1517	0.1445	0.1054	0.1974	0.12138
NCL Lin et al. [2022]	0.1121	0.1377	0.1576	0.1439	0.1272	0.1384	0.181	0.1535	0.1384	0.1005	0.1867	0.1152
BUIR Lee et al. [2021]	0.0640	0.0736	0.1036	0.0824	0.0842	0.0940	0.1216	0.1040	0.0551	0.0373	0.0830	0.0458
SSL4Rec Yao et al. [2021]	0.0811	0.0849	0.1142	0.0926	0.0576	0.0508	0.0958	0.0649	0.1491	0.1152	0.1924	0.1283
SelfCF Zhou et al. [2023]	0.0595	0.0662	0.0944	0.0741	0.0798	0.0909	0.1146	0.0998	0.0403	0.0269	0.0642	0.0341
DirectAU Wang et al. [2022]	0.0999	0.1136	0.1365	0.1197	0.1091	0.1144	0.1584	0.1295	0.1225	0.0882	0.1757	0.1041
SimGCL Yu et al. [2022]	0.1218	0.1470	0.1731	0.1540	0.1279	0.1391	0.1823	0.1544	0.1449	0.1067	0.1967	0.1222
DGCL	**0.1292**	**0.1593**	**0.1782**	**0.1639**	**0.1307**	**0.1424**	**0.1855**	**0.1577**	**0.1495**	**0.1090**	**0.2052**	**0.1259**

Table 2: Ablation study of DGCL, DGCL-w/o diff denotes the model variant without diffusion augmentation, and DGCL-w/o neg represents the variant without negative sampling.

Models	Douban-Book				Gowalla				Amazon-Kindle			
	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
DGCL - w/o diff	0.1246	0.1562	0.1740	0.1574	0.1294	0.1413	0.1845	0.1486	0.1486	0.1082	0.2042	0.1250
DGCL - w/o neg	0.0796	0.0928	0.1251	0.1015	0.0896	0.0994	0.1411	0.1223	0.0629	0.0784	0.1643	0.1462
DGCL	**0.1292**	**0.1593**	**0.1782**	**0.1639**	**0.1307**	**0.1424**	**0.1855**	**0.1577**	**0.1495**	**0.1090**	**0.2052**	**0.1259**

5.2 Experimental Results

As summarized in Table 1, DGCL consistently outperforms baseline methods across three public collaborative filtering datasets, demonstrating the efficacy of its diffusion-based augmentation. Notably, on Douban-Book, it improves N@10 and N@20 by 1.23% and 0.99% over SimGCL, and surpasses SimGCL by 0.85% in R@20 on Amazon-Kindle. Unlike uniform noise that introduces uncorrelated perturbations and degrades semantic coherence, DGCL employs an iterative denoising process to generate feature-adaptive augmentations that preserve semantic correlations and diversify contrastive views. Furthermore, SGL suffers from structural degradation due to random edge or node dropout, while NCL is limited by prototype quality and clustering reliability—especially in heterogeneous datasets like Amazon-Kindle. In contrast, DGCL flexibly produces diverse augmentations without relying on clustering, maintaining topological structure and uncovering underrepresented features through progressive generation. Thus, DGCL effectively avoids semantic deviation and enhances representation learning. A limitation of this work, similar to traditional diffusion algorithms, is the high computational cost during training and inference.

5.3 Ablation Study

In this section, we perform ablation studies to assess two key modules: negative sampling and diffusion augmentation. Results on three datasets (Table 2) show that removing diffusion augmentation (DGCL-w/o diff) causes noticeable declines, e.g., 0.31% in N@10 and 0.65% in N@20 on Douban-Book, demonstrating its ability to uncover latent representations and enrich feature semantics through diverse sample generation. Removing negative sampling (DGCL-w/o neg) leads to substantial drops across all metrics, underscoring its critical role in providing hard negatives that enhance discriminative power and complement diffusion. Both modules are pivotal to collaborative filtering and exhibit a synergistic effect.

[2]http://snap.stanford.edu/data/loc-gowalla.html

6 Conclusion

This paper proposes DGCL, a Diffusion-augmented Graph Contrastive Learning framework for collaborative filtering. DGCL integrates diffusion processes with graph contrastive learning to enhance recommendation through improved data augmentation. The model employs a forward noise-injection and reverse denoising process via a transformer-based encoder, recovering discriminative embeddings while maintaining graph topology. Augmented views are generated from estimated Gaussian parameters, and combined with negative sampling to achieve state-of-the-art results across three public datasets. DGCL effectively preserves semantic consistency and node-specific features, while exploring underrepresented regions of the feature space to enrich contrastive view diversity.

References

Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009(1):421425, 2009.

Yehuda Koren, Steffen Rendle, and Robert Bell. Advances in collaborative filtering. *Recommender systems handbook*, pages 91–142, 2021.

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

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. 2016.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. Self-supervised learning for recommender systems: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 36 (1):335–355, 2023.

Yangqin Jiang, Chao Huang, and Lianghao Huang. Adaptive graph contrastive learning for recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 4252–4261, 2023.

Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1294–1303, 2022.

Yonghui Yang, Zhengwei Wu, Le Wu, Kun Zhang, Richang Hong, Zhiqiang Zhang, Jun Zhou, and Meng Wang. Generative-contrastive graph learning for recommendation. In *Proceedings of the 46th international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1117–1126, 2023.

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

Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. Towards representation alignment and uniformity in collaborative filtering. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 1816–1825, 2022.

Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 726–735, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. Diffusion recommender model. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 832–841, 2023.

Yu Hou, Jin-Duk Park, and Won-Yong Shin. Collaborative filtering based on diffusion models: Unveiling the potential of high-order connectivity. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1360–1369, 2024.

Jeongwhan Choi, Seoyoung Hong, Noseong Park, and Sung-Bae Cho. Blurring-sharpening process models for collaborative filtering. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 1096–1106, 2023.

Yangqin Jiang, Yuhao Yang, Lianghao Xia, and Chao Huang. Diffkg: Knowledge graph diffusion model for recommendation. In *Proceedings of the 17th ACM international conference on web search and data mining*, pages 313–321, 2024.

Yunqin Zhu, Chao Wang, Qi Zhang, and Hui Xiong. Graph signal diffusion model for collaborative filtering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1380–1390, 2024.

Chaejeong Lee, Jeongwhan Choi, Hyowon Wi, Sung-Bae Cho, and Noseong Park. Scone: A novel stochastic sampling to generate contrastive views and hard negative samples for recommendation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 419–428, 2025.

Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. Mixgcf: An improved training method for graph neural network-based recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 665–674, 2021.

Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Dongha Lee, SeongKu Kang, Hyunjun Ju, Chanyoung Park, and Hwanjo Yu. Bootstrapping user and item representations for one-class collaborative filtering. In *Proceedings of the 44th international ACM SIGIR conference on Research and Development in information retrieval*, pages 317–326, 2021.

Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 4321–4330, 2021.

Xin Zhou, Aixin Sun, Yong Liu, Jie Zhang, and Chunyan Miao. Selfcf: A simple framework for self-supervised collaborative filtering. *ACM Transactions on Recommender Systems*, 1(2):1–25, 2023.

Trung-Kien Nguyen and Yuan Fang. Diffusion-based negative sampling on graphs for link prediction. In *Proceedings of the ACM Web Conference 2024*, pages 948–958, 2024.

Appendix

A pseudo-code of DGCL

In this section, we summarize the training and inference of DGCL and provide pseudo-code in detail.

Algorithm 1 Algorithm of DGCL Training

Input: user and item embedding E and randomly initialized θ.
Output: optimized θ.

```
Sample a batch of node embedding e ∈ E.
while converged do
    Sample t ∼ U(1, T) , ϵ ∼ N(1, I);
    Compute the noised embedding e_t given the e_0,ϵ via Eq. 3;
    Predict the noise from e_t iteratively by Eq. 4;
    Calculate the loss L_diff according to the Eq. 9;
    Take the gradient descent step on ∇_θ L_t to optimize θ;
end while
```

Algorithm 2 Algorithm of DGCL Inference

Input: embedding prediction model f_θ from the diffusion process, node embedding. e_0
Output: the augmented contrastive view embedding $\widetilde{e}$, including the user contrastive view $\widetilde{e}_u$ and item contrastive view $\widetilde{e}_i$.

```
Sample Gaussian noise z ∈ N(0, I).
Compute the initial noised data e_t in Eq. 3.
for t = T, ..., 1 do
    Calculate the μ_θ(e_t, t) and e_{t−1} via Eq. 11 and Eq. 12.
end for
```

B Theoretical Analysis

B.1 Manifold Reconstruction

theorem B.1 (Manifold Reconstruction)**.** *Suppose the reverse model satisfies* $p_\theta(e_{t-1} \mid e_t) = q(e_{t-1} \mid e_t, e_0)$ *for all* t *and that the prior used at* $t = T$ *equals* $q(e_T)$*. Then the marginal distribution of the reconstructed samples equals the data distribution:*$p_\theta(\hat{e}_0) = p_{data}(e_0)$*. Consequently, the support of* $p_\theta(\hat{e}_0)$ *coincides with the semantic manifold* $\mathcal{M}$*.*

Proof. Under the forward chain we have the joint

$$q(e_0, \ldots, e_T) = p_{\text{data}}(e_0) \prod_{t=1}^{T} q(e_t \mid e_{t-1}). \tag{16}$$

The reverse process is defined by:

$$p_\theta(e_0, \ldots, e_T) = p_\theta(e_T) \prod_{t=T}^{1} p_\theta(e_{t-1} \mid e_t). \tag{17}$$

By assumption, for every t we have $p_\theta(e_{t-1} \mid e_t) = q(e_{t-1} \mid e_t, e_0)$ (which depends implicitly on e_0 along the forward path) and $p_\theta(e_T) = q(e_T)$. The well-known time-reversal identity for Markov chains shows that these equalities imply $p_\theta(e_0, e_T) = q(e_0, e_T)$ for all (e_0, e_T) and that the full joint also matches:

$$p_\theta(e_0, \ldots, e_T) = q(e_0, \ldots, e_T). \tag{18}$$

Marginalizing out $e_1, \ldots, e_T$ from both sides yields

$$p_\theta(e_0) = \int p_\theta(e_{0:T})\, de_{1:T} = \int q(e_{0:T})\, de_{1:T} = q(e_0) = p_{\text{data}}(e_0). \tag{19}$$

Hence, the reconstructed samples $\hat{e}_0 \sim p_\theta$ are distributed according to the true data distribution. Therefore, the support of $p_\theta(\hat{e}_0)$ is exactly the semantic manifold $\mathcal{M}$, and DGCL approximately reconstructs this manifold in practice, supporting the capability of DGCL to approximately reconstructs the semantic manifold of original embeddings.

B.2 Semantic Consistency

theorem B.2 (Semantic Consistency)**.** *Under the assumed forward process and with the reverse model matching the true posterior, the sampled augmented embedding $\hat{e}_0$ satisfies $\mathbb{E}[\hat{e}_0 \mid e_0] = e_0$ i.e. the reverse-sampled view is mean-unbiased for the original embedding. Moreover, the conditional mean-squared error equals the posterior variance trace: $\mathbb{E}\big[\|\hat{e_0} - e_0\|^2 \mid e_0\big] = \mathrm{Tr}\big(\mathrm{Var}(e_0 \mid e_0)\big)$ The posterior variance can be bounded in terms of the diffusion schedule β_t; in particular there exists a constant $C(d)$ such that $\mathbb{E}\left[\|\hat{e}_0 - e_0\|^2\right] \le C(d) \sum_{t=1}^{T} \beta_t$.*

Proof. We prove mean-unbiasedness by conditioning on the forward–reverse path. Consider the forward marginal $q(e_t \mid e_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t} e_0, (1 - \bar{\alpha}_t) I)$. The true posterior at step t has the form (standard Gaussian conditioning): $q(e_{t-1} \mid e_t, e_0) = \mathcal{N}\left(e_{t-1}; \tilde{\mu}_t(e_t, e_0), \tilde{\beta}_t I\right)$, where the posterior mean can be written as a linear combination

$$\tilde{\mu}_t(e_t, e_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} e_0 + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} e_t. \tag{20}$$

Let $a_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}, \quad b_t = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$. Then:

$$\tilde{\mu}_t(e_t, e_0) = a_t e_0 + b_t e_t, \tag{21}$$

Taking expectation over $e_t | e_0$:

$$\mathbb{E}\big[\tilde{\mu}_t(e_t, e_0) \mid e_0\big] = a_t e_0 + b_t, \tag{22}$$

$$\mathbb{E}[e_t \mid e_0] = a_t e_0 + b_t \sqrt{\bar{\alpha}_t}, e_0 = (a_t + b_t \sqrt{\bar{\alpha}_t}) e_0. \tag{23}$$

A direct computation shows:

$$a_t + b_t \sqrt{\bar{\alpha}_t} = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t + \sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\sqrt{\bar{\alpha}_t}}{1 - \bar{\alpha}_t}. \tag{24}$$

Hence:

$$\mathbb{E}[e_{t-1} \mid e_0] = \mathbb{E}[\tilde{\mu}_t(e_t, e_0) \mid e_0] = \sqrt{\bar{\alpha}_{t-1}} e_0. \tag{25}$$

By induction, running the reverse chain from $t = T$ down to $t = 0$ and taking expectations stepwise yields

$$\mathbb{E}[\hat{e}_0 \mid e_0] = e_0. \tag{26}$$

The conditional MSE equals the posterior variance trace. The posterior variance accumulates contributions from sampling noise injected at each reverse step. For the MSE:

$$\mathbb{E}[\|\hat{e}_0 - e_0\|^2 \mid e_0] = \mathrm{Tr}\left(\mathrm{Var}(\hat{e}_0 \mid e_0)\right) + \|\mathbb{E}[\hat{e}_0 \mid e_0] - e_0\|^2. \tag{27}$$

Since the estimator is unbiased, the bias term vanishes and :

$$\mathbb{E}\left[\|\hat{e}_0 - e_0\|^2 \mid e_0\right] = \mathrm{Tr}\left(\mathrm{Var}(\hat{e}_0 \mid e_0)\right). \tag{28}$$

The variance at each reverse step is proportional to β_t, and by propagating variance through the chain, we obtain:

$$\mathrm{Var}(\hat{e}_0 \mid e_0) \preceq C(d) \sum_{t=1}^{T} \beta_t \cdot \mathbf{I}, \tag{29}$$

for some constant $C(d)$ depending on dimension d. Therefore: $\mathbb{E}\left[\|\hat{e}_0 - e_0\|^2\right] \leq C(d) \sum_{t=1}^{T} \beta_t$. This shows that the reconstruction error is controlled by the cumulative noise schedule, ensuring semantic consistency.

B.3 Controlled Diversity

proposition B.3. *Let $\hat{e}_0^{(1)}$ and $\hat{e}_0^{(2)}$ be two independent samples drawn from the reverse model conditioned on the same e_0. Then $\mathbb{E}\left[\|\hat{e}_0^{(1)} - \hat{e}_0^{(2)}\|^2 \mid e_0\right] = 2\,Tr\left(Var(\hat{e}_0 \mid e_0)\right)$. In particular, when the posterior variance is nonzero the reverse process produces distinct views in expectation, and the magnitude of diversity is controlled by the posterior covariance. Because DGCL estimates node-adaptive posterior covariances, diversity is node-adaptive.*

Proof. Since $\hat{e}_0^{(1)}$ and $\hat{e}_0^{(2)}$ are i.i.d. given e_0, we have:

$$\mathbb{E}\big[|\hat{e}_0^{(1)} - \hat{e}_0^{(2)}|^2 \mid e_0\big] = \mathbb{E}\big[|\hat{e}_0^{(1)}|^2 + |\hat{e}_0^{(2)}|^2 - 2\langle \hat{e}_0^{(1)}, \hat{e}_0^{(2)} \rangle \mid e_0\big] \tag{30}$$

This simplifies to :

$$2\mathbb{E}[\|\hat{e}_0\|^2 \mid e_0] - 2\|\mathbb{E}[\hat{e}_0 \mid e_0]\|^2 = 2\,\mathrm{Tr}\left(\mathrm{Var}(\hat{e}_0 \mid e_0)\right), \tag{31}$$

where we used the identity:

$$\mathbb{E}[\|\hat{e}_0\|^2] = \|\mathbb{E}[\hat{e}_0]\|^2 + \mathrm{Tr}(\mathrm{Var}(\hat{e}_0)). \tag{32}$$

Thus, the expected squared distance between two augmented views is proportional to the trace of the posterior covariance. Since the reverse process in DGCL learns node-specific variances, the diversity of augmented views is adaptive to each node's local feature distribution. Moreover, because the noise is Gaussian and centered at e_0, the diversity is semantically controlled and explores the local neighborhood of the true embedding.

C Experiments Detail And Analysis

Table 3: DGCL performance on different graph inference layer L.

Models	Douban-Book				Gowalla				Amazon-Kindle			
	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
DGCL layer=1	0.1269	0.1545	0.1742	0.1588	0.1287	0.1402	0.1818	0.1549	0.1488	0.1002	0.2023	0.1243
DGCL layer=2	0.1279	0.1583	0.1777	0.1633	0.1296	0.1416	0.1849	0.1571	0.1494	0.1090	0.2051	0.1258
DGCL layer=3	**0.1292**	**0.1593**	**0.1782**	**0.1638**	**0.1307**	**0.1424**	**0.1855**	**0.1577**	**0.1495**	**0.1090**	**0.2052**	**0.1259**

Experimental Settings. the LighGCN is employed as the basic recommendation embedding. The hidden dimension and learning rate of the DGCL are searched from {64, 128, 256, 512, 1024} and{1e-2, 1e-3, 4e-4, 1e-4}, respectively. The number of GNN layers is selected from {1,2,3}. λ is searched in { 0.01, 0.2, 0.3} and timestep of diffusion is searched in {10,20,30,50}. The noise β is tined in range of {1e-5, 2e-2}. In performance metrics, we adopt the widely used ranking metrics to evaluate the model, including the Recall@K (R@K) and the NDCG@K (N@K), where $K \in \{10, 20\}$. All experiments were conducted on a high-performance computing cluster equipped with NVIDIA A800 GPUs, each with 80GB of memory.

C.1 Hyper-Parameter Sensitivity Analysis

Effect of Graph Layer Depth L. To investigate the impact of graph layer depth, we vary L within the range {1,2,3}. As shown in Table 3, the model achieves optimal performance with $L = 3$ across all three datasets. It is demonstrated the shallow graph layers can not capture the high-order neighbor interactions and semantic dependencies. We are not increasing the layers because too many layers risk over-smoothing in GNN learning, where node embeddings become indistinguishable due to excessive aggregation. **Effect of Contrastive Loss λ.** As illustrated in Figure 2(a),(d), We evaluate the influence

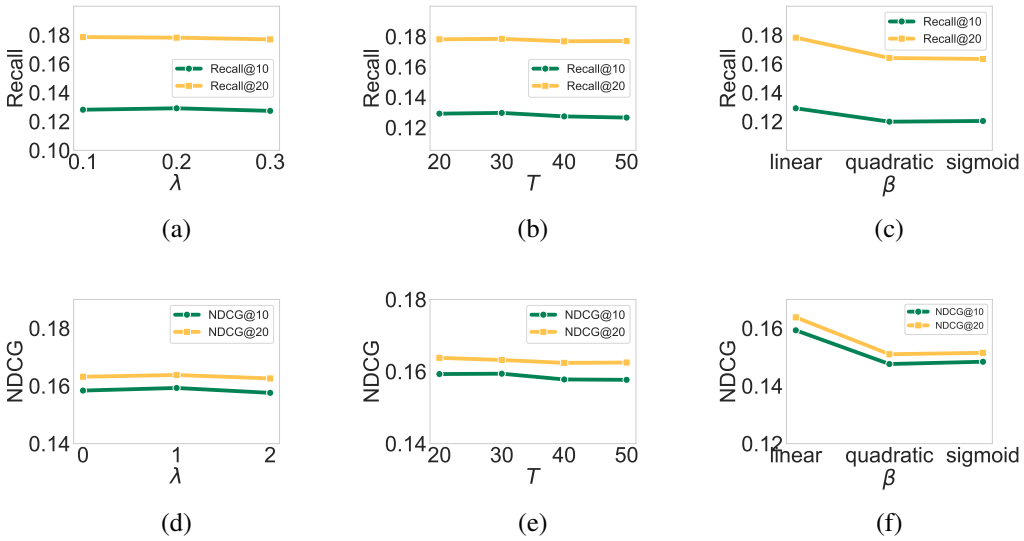


Figure 2: Effect of the λ, diffusion step T and noise β on Douban-Book

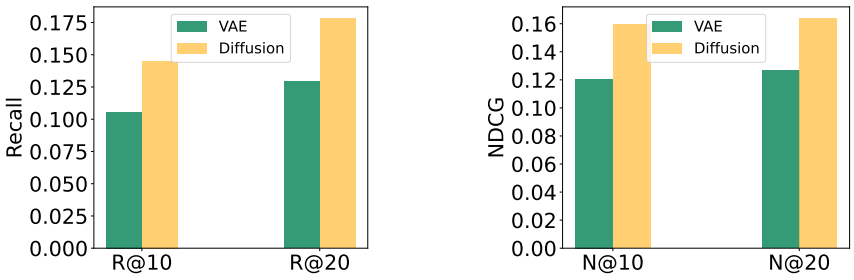
of the contrastive loss weight λ over{0.1,0.2,0.3} on the Douban-Book dataset. The results reveal that DGCL yields peak performance when λ is 0.2. A balanced weighting $(\lambda = 0.2)$ optimally integrates these objectives. High loss weight $(\lambda = 0.3)$ overemphasizes contrastive regularization, diluting task-specific signals, while lower value $(\lambda = 0.1)$ underutilizes the benefits of contrastive learning. Therefore, the recommendation loss plays a dominant role and drives task-specific learning, and the contrastive enhancement loss function serves as an auxiliary component which enhances embedding robustness by promoting invariance to augmented contrastive views. The integration of these two loss functions and joint training can improve the performance of the recommendation task.

Effect of Diffusion Step T. The diffusion step T critically governs the augmentation process by balancing noise injection and feature preservation. As illustrated in Figure 2(b)(e), the results indicate that the metrics obtain superior results when T is 30, with NDCG@20 slightly declining when T is 30. And as the diffusion step increases over time, the results tend to decrease gradually. This phenomenon may result from the multiple iterations of noise injection which may lead to excessive feature smoothing and the inability to capture the unique feature of each node. Moreover, more diffusion steps lead to more time cost and diversity loss. Therefore, sufficient steps are necessary to refine embeddings, but excess iterations harm discriminative power.

Effect of the Noise Schedule β. In this section, we evaluate three noise scheduling strategies for β: linear, quadratic, and sigmoid interpolation methods Nguyen and Fang [2024]. As shown in Figure 2(c), (f), the linear schedule outperforms alternatives. This may attributed to that sable diffusion facilitates the balance of noise interference and semantic coherence. In contrast, non-linear schedules (quadratic, sigmoid) disrupt the balance between perturbation and stability, leading to suboptimal augmentation.

C.2 Diffusion Augmentation Analysis

To validate the superior augmentation capability of our diffusion-based approach, we conduct a comparative analysis using a Variational Autoencoder (VAE)-based generation. Specifically, we



(a) (b)

Figure 3: Contrastive augmentation performance between Diffusion and VAE on Douban-Book.

replace the diffusion module in DGCL with a VAE and evaluate both methods on the Douban-Book dataset. As illustrated in Figure 3, the diffusion-augmented method consistently outperforms its VAE-augmented counterpart. This indicates that the multi-step denoising mechanism has advantages over VAE single-step reconstruction. The diffusion process employs iterative denoising steps to refine augmented samples progressively. Unlike VAE single-step augmentation, this gradual correction enables a deeper exploration of latent feature correlations, enhancing the model's ability to capture complex user-item interactions. Moreover, through implicit probabilistic modeling, the diffusion mechanism dynamically adjusts augmentation intensity based on local data density. DGCL can preserve semantic consistency with subtle perturbations in high-density regions and synthetic meaningful samples in low-density regions, thereby mitigating the data sparsity.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See Section Abstract and Introduction 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer:[Yes]

   Justification:See in Section Experimental Results 5.2. In the last sentence, we introduce the limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification:See theorem in section Theoretical Analysis 4 and proof in Appendix B

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See in Section Experiments 5 and Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is available in section Abstract or directly accessed at `https://anonymous.4open.science/r/DGCL-7FEA.`. Data is at section Experimental Setup 5.1.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See in Section Appendix Experimental Settings C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our paper includes experiments, but we have not reported error bars or other appropriate information about the statistical significance of the experiments. We acknowledge the importance of providing this information to support the robustness and reliability of our results. In future revisions, we will include error bars and conduct statistical significance tests to ensure that our findings are properly validated.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See in Section Appendix Experimental Settings C

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully reviewed the NeurIPS Code of Ethics and confirm that our research conforms to these guidelines in every respect. Our study adheres to the principles of research integrity, data handling, fairness, transparency, and consideration of societal impacts as outlined in the Code of Ethics. We have ensured that no ethical standards were violated during the research process and have accurately reported all relevant information in the paper.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification:Our work is foundational research focused on recommendation system. It is not directly tied to any specific applications or deployments that would have immediate societal impacts.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work does not involve the release of data or models that pose a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly credited the creators or original owners of all assets used in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not introduce any new assets such as datasets, models, or code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve any crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve any crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [No]

    Justification: We used LLMs in our research, but only for editing and formatting purposes

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.