Relation-aware Diffusion-Asymmetric Graph Contrastive Learning for Recommendation

Anonymous ACL submission

Abstract

Collaborative filtering (CF) based recommendation has been significantly enhanced by Graph Neural Networks (GNNs) and Graph Contrastive Learning (GCL), yet two persistent challenges remain: (i) random edge perturbations often destroy vital structural signals, degrading semantic consistency across augmented views; and (ii) data sparsity undermines generalization by limiting the propagation of collaborative signals. To address these issues, we propose **R**elation-**a**ware **D**iffusion-**A**symmetric Graph Contrastive Learning for **R**ecommendation (**RaDAR**), a novel contrastive framework that integrates two complementary view generation strategies: a graph generative model and a relation-aware graph denoising model. RaDAR introduces three key innovations: (1) *asymmetric contrastive learning* with global negative sampling to preserve semantic consistency while reducing noise; (2) *diffusion-guided augmentation*, which improves robustness through progressive noise injection and denoising; and (3) *relation-aware edge refinement*, which dynamically adjusts edge weights based on latent node semantics. Extensive experiments on three public benchmarks show that RaDAR consistently outperforms state-of-the-art recommendation methods, especially under noisy and sparse settings. The code of our method is available at our repository[1].

1 Introduction

Recommender systems(Wu et al., 2022) play a vital role in alleviating information overload by learning personalized preferences from sparse user-item interactions. A prevailing approach to recommendation is collaborative filtering (CF)(Schafer et al., 2007), which infers user interests based on historical behavioral patterns. To capture high-order connectivity and structural semantics, recent methods have leveraged Graph Neural Networks (GNNs) (Scarselli et al., 2008), which model user-item interactions through message passing on bipartite graphs. These advances have significantly improved recommendation accuracy, particularly in sparse settings.

To further enhance representation learning, Graph Contrastive Learning (GCL)(You et al., 2020) has emerged as a self-supervised paradigm that encourages consistency across multiple augmented views of the interaction graph. By integrating GCL with GNNs, recent models aim to improve robustness against data sparsity and noise. Typical implementations, such as SGL (Wu et al., 2021), generate graph augmentations through node or edge dropout, while methods like GraphACL (Xiao et al., 2024) introduce asymmetric contrastive objectives to capture multi-hop patterns. In parallel, diffusion-based models (Ho et al., 2020; Li et al., 2022) have shown promise in improving denoising capacity through iterative noise injection and reconstruction.

Despite these advancements, two fundamental challenges limit current GCL-based recommendation models: **Challenge 1 (C1): Structural Semantics Degradation.** Standard graph augmentations (e.g., random node/edge dropout) often corrupt essential topological structures, degrading collaborative signals and destabilizing contrastive learning. This structural perturbation compromises semantic consistency between augmented views, hindering effective representation learning. **Challenge 2 (C2): Limited Relational Expressiveness.** Existing methods predominantly assume homophily, emphasizing one-hop neighborhood alignment. However, real-world user interactions frequently exhibit heterophily or distant homophily—where similar users connect through multi-hop paths with weak direct links. Current models inadequately capture these higher-order relational patterns. While diffusion models en-

[1] https://anonymous.4open.science/r/RadarGCL-DB7B

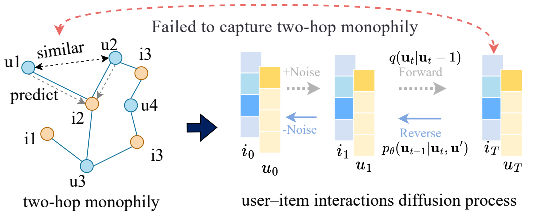


Figure 1: Illustration of the ACL mechanism and diffusion model for User-Item interaction graph: highlighting how conventional diffusion processes fail to capture crucial two-hop monophily patterns, where indirectly connected users (u1-u2) share similar preferences that are not reflected in simple user-item interaction diffusion.

hance noise robustness, they sacrifice fine-grained relational semantics beyond immediate neighborhoods. As illustrated in Fig 1, two-hop neighbors often share implicit preferences despite weak direct connections, which cannot be adequately modeled through conventional approaches.

To address recommendation challenges in sparse and noisy scenarios, we propose **RaDAR** (Relation-aware Diffusion-Asymmetric Graph Contrastive Learning for Recommendation), a contrastive learning framework with two core objectives: preserving structural semantics and enhancing relational expressiveness.

For C1 (structural semantics degradation), RaDAR introduces a *diffusion-guided augmentation strategy* applying Gaussian noise to node representations with learned denoising. This maintains semantic integrity while generating robust graph views for contrastive learning, reducing overfitting to spurious patterns.

For C2 (limited relational expressiveness), RaDAR employs a *dual-view generation architecture* combining: (i) a graph generative module based on variational autoencoders, capturing global structural semantics beyond one-hop connections; and (ii) a relation-aware graph denoising module that adaptively reweights edge contributions, preserving fine-grained relational signals. Additionally, RaDAR's *asymmetric contrastive objective* decouples node identity from structural context, enabling alignment of semantically similar nodes even in heterogeneous neighborhoods. Experiments on Last.FM, Yelp, and BeerAdvocate benchmarks show that RaDAR consistently outperforms 16 state-of-the-art baselines, especially under high sparsity and noise conditions.

In summary, our contributions are summarized as follows:

- We propose a novel dual-view graph contrastive framework that integrates diffusion-based augmentation and relation-aware graph denoising;
- We introduce a unified optimization scheme combining asymmetric contrastive learning with noise-resilient diffusion to preserve multi-hop semantics;
- We achieve new state-of-the-art results on multiple recommendation benchmarks, with consistent gains under both clean and noisy interaction scenarios.

2 Preliminaries and Related Work

2.1 Collaborative Filtering Paradigm

Let U and V denote user and item sets, with interactions encoded in a binary matrix. Graph-based collaborative filtering extracts representations by propagating information across the interaction graph under the homophily principle: users with similar interaction patterns share preferences. Implementations typically employ dual-tower architectures to map users and items into a shared latent space, enabling relevance estimation through similarity matching. This approach captures transitive dependencies in interaction graphs to infer unobserved user-item affinities.

2.2 Self-Supervised Graph Learning

Recent advances in graph neural networks (GNNs) have revolutionized recommendation systems through structured modeling of user-item interactions. Core architectures including PinSage(Ying et al., 2018), NGCF(Wang et al., 2019), and LightGCN(He et al., 2020) employ graph convolution operations to encode multi-hop relational patterns, with LightGCN achieving computational efficiency through neighbor aggregation simplification. Subsequent refinements integrate multi-intent disentanglement (DGCF(Wang et al., 2020a), DCCF(Ren et al., 2023)) and adaptive relation discovery (DRAN(Wang et al., 2022b)) to enhance representation learning. Temporal dynamics are further captured through graph-enhanced sequence modeling (DGSR(Zhang et al., 2022), GCE-GNN(Wang et al., 2020b)) that bridges historical interactions with evolving preferences.

The integration of self-supervised learning (SSL) with graph techniques has emerged as a paradigm

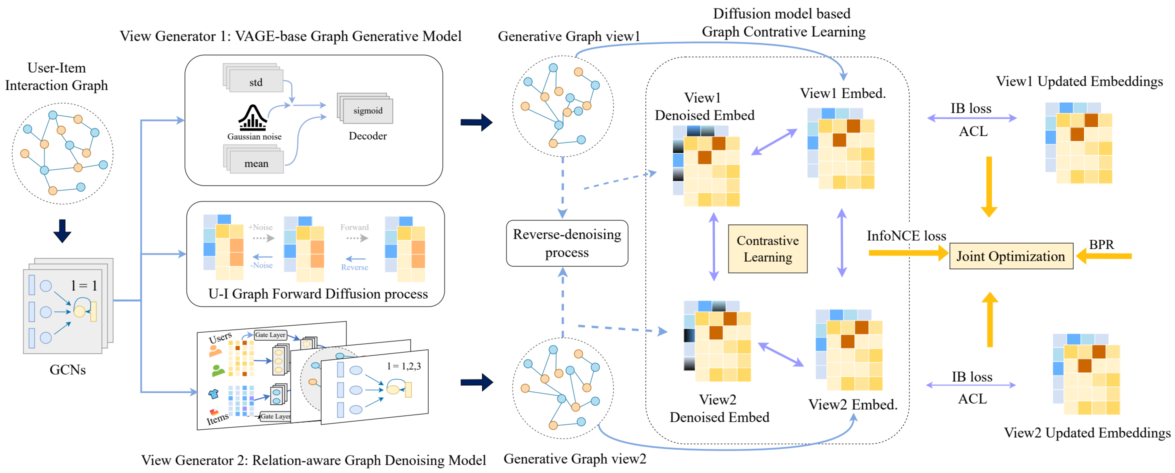


Figure 2: RaDAR framework architecture: The left section shows two view generators extracting complementary graph representations. The right section demonstrates the contrastive learning process with diffusion model-based graph generation and joint optimization through InfoNCE, IB, and BPR losses.

for data-efficient representation learning. Contrastive frameworks like SGL(Wu et al., 2021) and GFormer(Li et al., 2023) construct augmented graph views to improve user-item embeddings through invariance learning, while reconstruction-based methods (S3-Rec(Zhou et al., 2020)) exploit masked interaction prediction. SSL has demonstrated cross-domain effectiveness through cross-view contrastive alignment (C2DSR(Cao et al., 2022)) and multi-modal pattern discovery (SLMRec(Tao et al., 2022)), establishing its versatility in addressing diverse recommendation challenges through auxiliary self-supervision signals.

3 Methodology

In this section, we present the comprehensive architecture of RaDAR, which consists of four interconnected components. The first component employs a graph message passing encoder to effectively capture local collaborative relationships between users and items. The second component implements a sophisticated user-item graph diffusion model. The third component integrates an adaptive framework featuring two distinct trainable view generators: one leveraging a graph variational model and another utilizing relation-aware denoising graph models. The fourth component focuses on model optimization through a multi-faceted loss function that incorporates ACL to boost performance, complemented by diffusion model-based Graph Contrastive Learning. The overall architecture of the RaDAR model is illustrated in Figure 2.

3.1 User-item Embedding Propagation

We project users and items into a d-dimensional latent space through learnable embeddings, denoted as $\mathbf{E}^{(u)} \in \mathbb{R}^{N \times d}$ and $\mathbf{E}^{(v)} \in \mathbb{R}^{M \times d}$ for N users and M items. To capture collaborative signals, we employ a normalized adjacency matrix derived from the interaction matrix (see Eq. 9 in Appendix B.1).

The embedding propagation process utilizes a multi-layer graph neural network where user and item representations are iteratively refined through message passing (Eq. 10 in Appendix B.1). The final embeddings integrate information across all L layers through summation (Eq. 11). We compute the preference score between user u_i and item v_j via inner product of their respective embeddings.

3.2 GCL Paradigm

3.2.1 Graph Generative Model as View Generator

We adopt Variational Graph Auto-Encoder (VGAE) (Kipf and Welling, 2016) for view generation, integrating variational inference with graph reconstruction. The encoder employs multi-layer GCN for node embeddings, while the decoder reconstructs graph structures using Gaussian-sampled embeddings. The VGAE framework optimizes a multi-component loss function comprising KL-divergence regularization (Eq. 13), discriminative loss for reconstructing graph structure (Eq. 14), and Bayesian Personalized Ranking loss (Eq. 15). The complete formulation of the VGAE objective is provided in Appendix B.2 (Eq. 16).

3.2.2 Relation-Aware Graph Denoising for View Generation

Our denoising framework employs layer-wise edge masking with sparsity constraints (Eq. 17). We model edge retention through reparameterized Bernoulli distributions with parameters learned via relation-aware denoising layers that employ adaptive gating (Eq. 18). The framework utilizes a GRU-inspired mechanism (Cho, 2014) for relational filtering and employs concrete distribution for differentiable edge sampling. The training objective combines concrete distribution regularization with recommendation loss (Eq. 21). The complete mathematical details are provided in AppendixB.3.

3.3 Diffusion with User-Item Graph

Building on diffusion models' noise-to-data generation capabilities(Wang et al., 2023; Ho et al., 2020; Sohl-Dickstein et al., 2015), we propose a graph diffusion framework that transforms the original user-item graph $\mathcal{G}_{ui}$ into recommendation-optimized subgraphs $\mathcal{G}'_{ui}$. We design a forward-inverse diffusion mechanism: forward noise injection gradually degrades node embeddings via Gaussian perturbations, while inverse denoising recovers semantic patterns through learned transitions. This process enhances robustness against interaction noise while learning complex embedding distributions. The restored embeddings produce probability distributions for subgraph reconstruction, establishing an effective diffusion paradigm for high-fidelity recommendation graph generation.

3.3.1 Noise Diffusion Process

Our framework introduces a latent diffusion paradigm for graph representation learning, operating on GCN-derived embeddings rather than graph structures. Let $\mathbf{h}^L$ denote the item embedding from the final GCN layer. We construct a T-step Markov chain $\chi_{0:T}$ with initial state $\chi_0 = \mathbf{h}_0^{(L)}$.

The forward process progressively adds Gaussian noise to embeddings, transforming them towards a standard normal distribution. Through reparameterization techniques (detailed in Appendix A.1), we can directly compute any intermediate state from the initial embedding:

$$\chi_t = \sqrt{\bar{\alpha}_t}\chi_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}) \tag{1}$$

To precisely control noise injection, we implement a linear noise scheduler with hyperparameters s, α_{low}, and α_{up} Appendix B.4.

The reverse process employs neural networks parameterized by θ to progressively denoise representations, recovering the original embeddings through learned Gaussian transitions. This denoising procedure enables our model to capture complex patterns in the graph-derived embeddings while maintaining their structural properties.

3.3.2 Diffusion Process Optimization for User-Item Interaction.

The optimization objective is formulated to maximize the Evidence Lower Bound (ELBO) of the item embedding likelihood χ_0. Following the diffusion framework in (Jiang et al., 2024), we derive the training objective as:

$$\mathcal{L}_{elbo} = \mathbb{E}_{t\sim\mathcal{U}(1,T)}\mathcal{L}_t. \tag{2}$$

where $\mathcal{L}_t$ denotes the loss at diffusion step t, computed by uniformly sampling timesteps during training. The ELBO comprises two components:(1)A reconstruction term $\mathbb{E}_{q(\chi_1|\chi_0)}\left[||\hat{\chi}_\theta(\chi_1, 1) - \chi_0||_2^2\right]$ that evaluates the model's denoising capability at $t = 1$, and (2)KL regularization terms governing the reverse process transitions. Following (Jiang et al., 2024), we minimize the KL divergence between the learned reverse distribution $p_\theta(\chi_{t-1}|\chi_t)$ and the tractable posterior $q(\chi_{t-1}|\chi_t$ The neural network $\hat{\chi}_\theta(\cdot)$, implemented as a Multi-Layer Perceptron (MLP), predicts the original embedding χ_0 from noisy embeddings χ_t and timestep encodings. This formulation preserves the theoretical guarantees of ELBO maximization.

3.4 Contrastive Learning paradigms

3.4.1 Diffusion-Enhanced Graph Contrastive Learning

We propose a diffusion-augmented contrastive framework leveraging intra-node self-discrimination for self-supervised learning. Given node embeddings $E^{'}$ and $E^{''}$ from two augmented views, we consider augmented views of the same node as positive pairs (e'_i, e''_i), and views of different nodes as negative pairs $(e'_i, e''_{i'})$ where $u_i \neq u_{i'}$. The formulation of the loss function is:

$$\mathcal{L}_{ssl}^{user} = \sum_{u_i \in \mathcal{U}} -\log \frac{\exp(s(e'_i, e''_i)/\tau)}{\sum_{u_j \in \mathcal{U}} \exp(s(e'_i, e''_j)/\tau)}, \tag{3}$$

where $s(\cdot)$ denotes cosine similarity and τ represents the temperature parameter. The item-side contrastive loss $\mathcal{L}_{ssl}^{item}$ follows an analogous formulation. The complete self-supervised objective combines both components:

$$\mathcal{L}_{ssl} = \mathcal{L}_{ssl}^{user} + \mathcal{L}_{ssl}^{item} \tag{4}$$

Our diffusion-enhanced augmentation generates denoised views ($\mathcal{V}_1^{den}$, $\mathcal{V}_2^{den}$) via Markov chains that preserve interaction patterns while suppressing high-frequency noise. The framework implements: (i) *Intra-View Alignment* (L_{intra}), which measures the contrastive loss between original view $\mathcal{V}_i$ and its denoised counterpart $\mathcal{V}_i^{den}$. (ii) *Inter-View Regularization* (L_{inter}), which computes the contrastive loss between different denoised views $\mathcal{V}_1^{den}$ and $\mathcal{V}_2^{den}$.

The composite loss integrates these mechanisms:

$$L_{\text{diff-ssl}} = L_{\text{ssl}} + \lambda_1 L_{\text{intra}} + \lambda_2 L_{\text{inter}} \tag{5}$$

where λ_1 and λ_2 balance view consistency and information diversity. This design enables simultaneous noise suppression and multi-perspective representation learning.

3.4.2 Asymmetric Graph Contrastive Learning

Conventional contrastive frameworks are limited by homophily assumptions (Lim et al., 2021; Chin et al., 2019). We adopt an asymmetric paradigm (Xiao et al., 2024) for monophily-structural contexts using dual encoders f_θ and f_ξ that generate identity and context representations. An asymmetric predictor reconstructs neighborhood contexts from node identities (Eq. 27 in Appendix B.5). This preserves node semantics while encoding structural patterns, naturally accommodating monophily through shared central nodes. Our dual-representation framework uses view-specific encoders f_θ and f_ξ to generate identity representations $\mathbf{v} = f_\theta(G)[\mathbf{v}]$ and context representations $\mathbf{u} = f_\xi(G)[\mathbf{u}]$. An asymmetric predictor g_ϕ reconstructs neighborhood contexts from node identities, optimizing a contrastive objective (see Eq. 27 in Appendix B.5).

This formulation achieves two key properties: (1) identity representations preserve node-specific semantics, and (2) context representations encode structural neighborhood patterns. The asymmetric objective naturally accommodates monophily by enabling two-hop neighbors to reconstruct similar contexts through their shared central nodes.

Table 1: Training phases in our framework.

Phase	Objective	Params
1	$L_{\text{bpr}}, L_{\text{diff-ssl}}, \|\Theta\|_F^2$	User-item embeds
2	$\mathcal{L}_{IB}$ (Info. bottleneck)	User-item embeds
3	$\mathcal{L}_{gen} + \mathcal{L}_{den}$	View generators

Table 2: Statistics of the experimental datasets.

Dataset	Users	Items	Interactions	Density
Last.FM	1,892	17,632	92,834	2.8×10^{-3}
Yelp	42,712	26,822	182,357	1.6×10^{-4}
BeerAdvocate	10,456	13,845	1,381,094	9.5×10^{-3}

3.5 Model Training

Our framework adopts a hierarchical optimization approach with three coupled stages, as summarized in Table 1.

Phase 1: Unified Multi-Task Learning We initiate joint optimization:

$$L_1 = L_{\text{bpr}} + \lambda_3 L_{\text{diff-ssl}} + \lambda_4 \|\Theta\|_F^2 \tag{6}$$

where $L_{\text{diff-ssl}}$ is the diffusion-based self-supervised loss from Eq. 5, and $\|\Theta\|_F^2$ is L2 regularization.

Phase 2: Representation Distillation We impose an information bottleneck constraint:

$$\begin{aligned} \mathcal{L}_{IB} &= L_A(G, g_\phi(\mathbf{v}), \mathbf{v}, \mathbf{u}) \\ &= L_A(G, g_\phi(\mathbf{y}^*), \mathbf{y}^*, \hat{\mathbf{y}}), \end{aligned} \tag{7}$$

where $\mathbf{y}^*$ represents historical representations and L_A is the ACL loss.

Phase 3: View Generator Optimization We finalize training by optimizing view generators:

$$\mathcal{L}_{generators} = \mathcal{L}_{gen} + \mathcal{L}_{den} \tag{8}$$

where $\mathcal{L}_{gen}$ is the VGAE graph generation loss(see Eq. 16) and $\mathcal{L}_{den}$(see Eq. 21) is the relation-aware denoising loss.

3.5.1 Evaluation Datasets

We evaluate our method on three publicly available datasets:

- **Last.FM**(Celma, 2010): Music listening behaviors and social interactions from Last.fm users.
- **Yelp** (Yelp, 2018): A benchmark dataset of user-business ratings from Yelp, widely utilized in location-based recommendation studies.
- **BeerAdvocate** (McAuley and Leskovec, 2013): Beer reviews from BeerAdvocate, preprocessed with 10-core filtering to ensure data density.

Table 3: Performance Metrics for Various Models

Model	Last.FM				Yelp				BeerAdvocate			
	Recall@20	NDCG@20	Recall@40	NDCG@40	Recall@20	NDCG@20	Recall@40	NDCG@40	Recall@20	NDCG@20	Recall@40	NDCG@40
BiasMF	0.1879	0.1362	0.2660	0.1653	0.0532	0.0264	0.0802	0.0321	0.0996	0.0856	0.1602	0.1016
NCF	0.1130	0.0795	0.1693	0.0952	0.0304	0.0143	0.0487	0.0187	0.0729	0.0654	0.1203	0.0754
AutoR	0.1518	0.1114	0.2174	0.1336	0.0491	0.0222	0.0692	0.0268	0.0816	0.0650	0.1325	0.0794
PinSage	0.1690	0.1228	0.2402	0.1472	0.0510	0.0245	0.0743	0.0315	0.0930	0.0816	0.1553	0.0980
STGCN	0.2067	0.1558	0.2940	0.1821	0.0562	0.0282	0.0856	0.0355	0.1003	0.0852	0.1650	0.1031
GCMC	0.2218	0.1714	0.3149	0.1897	0.0584	0.0280	0.0891	0.0360	0.1082	0.0901	0.1766	0.1085
NGCF	0.2081	0.1474	0.2944	0.1829	0.0681	0.0336	0.1019	0.0419	0.1033	0.0873	0.1653	0.1032
GCCF	0.2222	0.1642	0.3083	0.1931	0.0724	0.0365	0.1151	0.0466	0.1035	0.0901	0.1662	0.1062
LightGCN	0.2349	0.1704	0.3220	0.2022	0.0761	0.0373	0.1175	0.0474	0.1102	0.0943	0.1757	0.1113
SLRec	0.1957	0.1442	0.2792	0.1737	0.0665	0.0327	0.1032	0.0418	0.1048	0.0881	0.1723	0.1068
NCL	0.2353	0.1715	0.3252	0.2033	0.0806	0.0402	0.1230	0.0505	0.1131	0.0971	0.1819	0.1150
SGL	0.2427	0.1761	0.3405	0.2104	0.0803	0.0398	0.1226	0.0502	0.1138	0.0959	0.1776	0.1122
HCCF	0.2410	0.1773	0.3232	0.2051	0.0789	0.0391	0.1210	0.0492	0.1156	0.0990	0.1847	0.1176
SHT	0.2420	0.1770	0.3235	0.2055	0.0794	0.0395	0.1217	0.0497	0.1150	0.0977	0.1799	0.1156
DirectAU	0.2422	0.1727	0.3356	0.2042	0.0818	0.0424	0.1226	0.0524	0.1182	0.0981	0.1797	0.1139
AdaGCL	0.2603	0.1911	0.3531	0.2204	0.0873	0.0439	0.1315	0.0548	0.1216	0.1015	0.1867	0.1182
Ours	**0.2724**	**0.1992**	**0.3664**	**0.2309**	**0.0914**	**0.0464**	**0.1355**	**0.0571**	**0.1273**	**0.1061**	**0.1942**	**0.1375**
Improv	4.65%	4.24%	3.77%	4.76%	4.70%	5.69%	3.04%	4.20%	4.69%	4.53%	4.02%	16.33%
p-val	$2.4e^{-6}$	$5.8e^{-5}$	$4.9e^{-9}$	$6.4e^{-5}$	$1.3e^{-4}$	$8.8e^{-4}$	$7.6e^{-3}$	$2.2e^{-3}$	$1.2e^{-4}$	$7.9e^{-4}$	$1.4e^{-4}$	$2.9e^{-6}$

4 Experimental Evaluation

To rigorously evaluate the proposed model, we design experiments to investigate four critical aspects:

- **RQ1**: How does *RaDAR* perform against state-of-the-art recommendation baselines in benchmark comparisons?
- **RQ2**: What is the individual contribution of key components to the model's effectiveness across diverse datasets? (Ablation Analysis)
- **RQ3**: How robust is *RaDAR* in handling data sparsity and noise compared to conventional approaches?
- **RQ4**: How do critical hyperparameters influence the model's performance characteristics?

4.1 Experimental Settings

4.1.1 Evaluation Protocols

Following standard evaluation protocols for recommendation systems, we partition datasets into training/validation/test sets (7:2:1). Adopting the all-ranking strategy, we evaluate each user by ranking all non-interacted items alongside test positives. Performance is measured using Recall@20 and NDCG@20 metrics, with N=20 as the default ranking cutoff. This setup ensures comprehensive assessment of model capabilities in real-world sparse interaction scenarios.

4.1.2 Compared Baseline Methods

We evaluate RaDAR against 16 representative baselines spanning four research streams: 1) **Traditional CF models**: BiasMF (Koren et al., 2009), NCF (He et al., 2017); 2) **GNN-based methods**: LightGCN (He et al., 2020), NGCF (Wang et al., 2019); 3) **Self-supervised frameworks**: SGL (Wu et al., 2021), SLRec (Yao et al., 2021); 4) **Contrastive learning**: DirectAU (Wang et al., 2022a), AdaGCL (Jiang et al., 2023). Full baseline descriptions and implementation details are provided in Appendix A. This taxonomy ensures coverage of both foundational approaches and cutting-edge paradigms, enabling rigorous evaluation across methodological dimensions.

Table 4: Ablation study on key components of RaDAR.

Model	Variant Description	Last.FM		Yelp		Beer	
		Recall	NDCG	Recall	NDCG	Recall	NDCG
Baseline	SOTA SSL	0.2603	0.1911	0.0873	0.0439	0.1216	0.1015
RaDAR	Gen+Gen	0.2665	0.1936	0.0900	0.0456	0.1226	0.1027
	Gen+Linear	0.2698	0.1986	0.0910	0.0461	0.1247	0.1050
	w/o D-ACL	0.2652	0.1934	0.0904	0.0458	0.1250	0.1036
RaDAR(full)		0.2724	0.1992	0.0914	0.0464	0.1273	0.1061

4.2 Overall Performance Comparison (RQ1)

Table 3 demonstrates RaDAR's superior performance across three benchmarks, outperforming existing methods in top-20/40 recommendations. This advantage derives from three key innovations: (1) relation-aware graph denoising that eliminates spurious correlations, (2) asymmetric contrastive learning preserving collaborative signals, and (3) diffusion-based iterative noise reduction. Unlike conventional approaches that compromise structural integrity through random augmentation, RaDAR's structural-preserving dual-view framework integrates noise-suppressed distribution modeling with relation-aware signal enhancement, effectively mitigating degradation in noisy interaction graphs.

4.3 Model Ablation Test (RQ2)

To evaluate RaDAR's architectural components, we conducted systematic ablation studies against the state-of-the-art baseline. We examined four configurations across three datasets (Last.FM, Yelp, and Beer):

- **RaDAR (Gen+Gen):** Dual VGAE-based generators without denoising model
- **RaDAR (Gen+Linear):** Linear attention replacing relation-aware denoising model
- **RaDAR (w/o D-ACL):** Conventional graph contrastive loss without diffusion-asymmetric contrastive learning optimization
- **RaDAR (full):** Complete proposed framework

Table 4 reveals critical performance differentials, demonstrating three key insights:

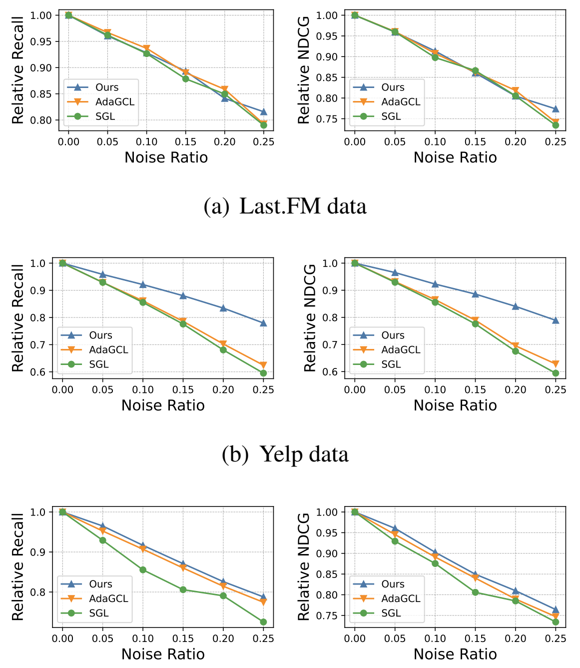
Relation-Aware Denoising Superiority: Our relation-aware denoiser demonstrates superior performance over alternatives. Substituting it with linear attention reduces Recall@20 by 0.95% (0.2724→0.2698), while VGAE generators yield a 2.17% decrease (0.2724→0.2665). This confirms enhanced noise-handling through explicit relation modeling compared to standard linear layers and VGAE architectures.

Diffusion-Asymmetric Contrastive Learning Synergy: While diffusion-based graph contrastive learning improves embedding robustness, asymmetric contrastive optimization captures multi-hop relationships through its loss formulation. Their synergistic integration enables noise-resilient embeddings while preserving relational patterns. Ablation studies confirm D-ACL's criticality, as its removal causes a 2.64% performance drop (Recall@20: 0.2724→0.2652), exceeding the impact of relation-aware denoising ablation (-2.17%).

Component Complementarity: The performance hierarchy (full > Gen+Linear > Gen+Gen > w/o D-ACL) reveals complementary mechanisms: relation-aware denoising eliminates noise through adaptive graph rewiring, while diffusion-ACL enhances contrastive effectiveness.

4.4 Model Robustness Test (RQ3)

In this section, our extensive experimental evaluation demonstrates the efficacy of our proposed RaDAR framework. The results indicate that RaDAR exhibits remarkable resilience against data noise and significantly outperforms existing methods in handling sparse user-item interaction data. Specifically, our approach maintains high performance even in the presence of substantial noise, showcasing its robust nature.



(a) Last.FM data

(b) Yelp data

(c) BeerAdvocate data

Figure 3: Impact of Noise Ratio (5%–25%) on Performance Degradation

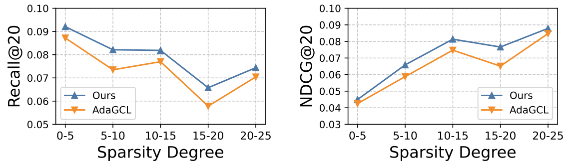
4.4.1 Performance w.r.t. Data Noise Degrees.

We systematically evaluate RaDAR's resilience to data corruption through controlled noise injection experiments, where spurious edges replace genuine interactions at incremental ratios (5%-25%). A comparative analysis with AdaGCL and SGL across datasets of varying density (Fig. 3) reveals two key patterns:

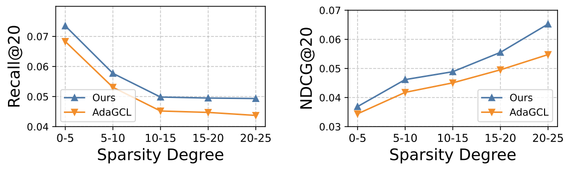
On moderate-density datasets (Last.FM: 2.8×10^{-3}, Beer: 9.5×10^{-3}), RaDAR demonstrates a modest improvement over AdaGCL on the Beer dataset, while the relative *Recall*/*NDCG* robustness performance among RaDAR, AdaGCL, and GCL shows less significant variation on the Last.FM dataset. This suggests that the benefits of our proposed approach may be less pronounced when data sparsity is moderate, as the existing methods already capture sufficient structural information under these conditions.

In extreme sparsity conditions (Yelp: 1.6×10^{-4}), RaDAR demonstrates pronounced advantage higher relative improvement margins, confirming superior noise resilience in data-scarce scenarios.

Our empirical analysis demonstrates RaDAR's effectiveness in cold-start scenarios through its density-aware denoising framework. The widening performance gap under increasing sparsity highlights the model's ability to extract critical signals from sparse interactions - a pivotal requirement for practical recommendation systems.



(a) Performance w.r.t. user interaction numbers



(b) Performance w.r.t. item interaction numbers

Figure 4: Performance analysis across five user and item interaction sparsity levels on Yelp dataset.

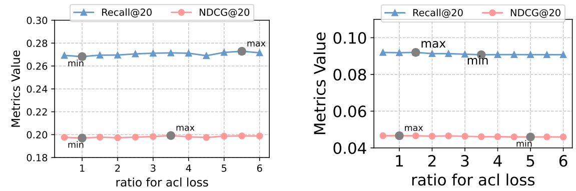
4.4.2 Performance w.r.t. Data Sparsity.

We analyze recommendation performance under varying interaction sparsity from dual user-item perspectives. As shown in Fig. 4(a), RaDAR exhibits marked superiority over AdaGCL across all user interaction groups, particularly in cold-start scenarios (0-10 interactions). This demonstrates its robustness in learning from sparse user behavior through adaptive graph augmentation. Contrastingly, the item-centric analysis (Fig. 4(b)) reveals an inverse trend: RaDAR's performance gap widens as item interaction density increases. This divergence highlights distinct sparsity-response items—user metrics generally degrade with sparsity (except minor recovery at 20-25 interactions), while item performance positively correlates with interaction frequency.

These results validate RaDAR's dual mechanisms: (1) Sparse user modeling via adaptive augmentation ensures stability in cold-start scenarios, and (2) Density-aware regularization captures item-side collaborative signals effectively. The opposing trends underscore RaDAR's balanced capability in addressing both user and item sparsity challenges.

4.5 Hyperparameter Analysis (RQ4)

We investigate the impact of the adjustable contrastive learning (ACL) ratio λ, which balances Information Bottleneck (IB) losses between the VAGE-base and relation-aware graph denoising view generators. The total IB loss is formulated as $L_{IB} = L_{IB}^{G} + \lambda L_{IB}^{D}$ where L_{IB}^{G} and L_{IB}^{D} represent the IB losses from the VAGE-base view generator and the relation-aware graph denoising view generator, and $\lambda > 1$ prioritizes relation-aware structural preservation, while $\lambda < 1$ emphasizes generated graph views.



(a) Last.FM (b) Yelp

Figure 5: Performance variation with ACL ratio λ. Last.FM peaks Recall@20 at $\lambda = 5.5$, NDCG@20 at $\lambda = 3.5$. Yelp peaks Recall@20 at $\lambda = 1.5$, NDCG@20 at $\lambda = 1.0$. Higher λ values enhance relation-aware denoising for Last.FM, while Yelp requires balanced contributions due to interaction sparsity.

Fig. 5 reveals distinct λ preferences across datasets. Last.FM achieves optimal performance with $\lambda > 1$(Fig. 5(a)), indicating its structural complexity benefits from enhanced relation-aware denoising. Conversely, Yelp attains peak metrics at lower λ values (Fig. 5(b)), suggesting its sparse interaction patterns require balanced information preservation from both view generators to prevent overfitting. This empirical evidence confirms RaDAR's adaptability through our symmetric contrastive learning design, demonstrating robust performance across diverse graph recommendation scenarios.

5 Conclusion

We propose RaDAR, a contrastive recommendation framework with three key innovations: (1) a dual-view architecture combining generative reconstruction and relation-aware denoising, (2) asymmetric contrastive learning for pattern discrimination, and (3) diffusion-based stabilization for robust feature learning. Experimental results demonstrate RaDAR's superior noise resilience compared with state-of-the-art baselines, with ablation studies validating the effectiveness of its graph purification and contrastive components. By explicitly separating collaborative signals from spurious correlations, our framework establishes principled design guidelines for contrastive recommenders. The methodology provides foundational insights for noise-resistant system development while maintaining interaction semantics, offering natural extensions to dynamic and multi-modal recommendation scenarios.

Limitations

This work has two key limitations: First, the current framework is restricted to homogeneous user-item graphs and lacks modality-specific components for cross-modal alignment and noise modeling in multi-modal scenarios. Second, its relational modeling capacity is constrained by single-relation bipartite graph assumptions, with limited capability to capture complex relational structures (e.g., multi-relational knowledge graphs) and relation-specific propagation patterns. Future work should explore modality-aware alignment mechanisms and relation-aware graph architectures to address these challenges.

References

Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*.

Jiangxia Cao, Xin Cong, Jiawei Sheng, Tingwen Liu, and Bin Wang. 2022. Contrastive cross-domain sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 138–147.

O. Celma. 2010. *Music Recommendation and Discovery in the Long Tail*. Springer.

Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 27–34.

Alex Chin, Yatong Chen, Kristen M. Altenburger, and Johan Ugander. 2019. Decoupled smoothing on graphs. In *The World Wide Web Conference*, pages 263–272.

Kyunghyun Cho. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. 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.

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*, pages 173–182.

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

Yangqin Jiang, Chao Huang, and Lianghao Huang. 2023. Adaptive graph contrastive learning for recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4252–4261.

Yangqin Jiang, Yuhao Yang, Lianghao Xia, and Chao Huang. 2024. Diffkg: Knowledge graph diffusion model for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 313–321.

Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Chaoliu Li, Lianghao Xia, Xubin Ren, Yaowen Ye, Yong Xu, and Chao Huang. 2023. Graph transformer for recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1680–1689.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. 2021. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems*, 34:20887–20902.

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*, pages 2320–2329.

Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*, pages 897–908.

Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. 2023. Disentangled contrastive collaborative filtering. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1137–1146.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web: methods and strategies of web personalization*, pages 291–324. Springer.

Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web*, pages 111–112.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265. PMLR.

Zhulin Tao, Xiaohao Liu, Yewei Xia, Xiang Wang, Lifang Yang, Xianglin Huang, and Tat-Seng Chua. 2022. Self-supervised learning for multimedia recommendation. *IEEE Transactions on Multimedia*, 25:5107–5116.

Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022a. 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.

Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. 2023. Diffusion recommender model. In *SIGIR*.

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

Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020a. Disentangled graph collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 1001–1010.

Zhaobo Wang, Yanmin Zhu, Haobing Liu, and Chunyang Wang. 2022b. Learning graph-based disentangled representations for next poi recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1154–1163.

Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. 2020b. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 169–178.

Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. 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.

Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37.

Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022a. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, pages 70–79.

Lianghao Xia, Chao Huang, and Chuxu Zhang. 2022b. Self-supervised hypergraph transformer for recommender systems. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2100–2109.

Teng Xiao, Huaisheng Zhu, Zhengyu Chen, and Suhang Wang. 2024. Simple and asymmetric graph contrastive learning without augmentations. *Advances in Neural Information Processing Systems*, 36.

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

Yelp. 2018. Yelp open dataset. https://business.yelp.com/data/resources/open-dataset/.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823.

Jiani Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. 2019. Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems. *arXiv preprint arXiv:1905.13129*.

Mengqi Zhang, Shu Wu, Xueli Yu, Qiang Liu, and Liang Wang. 2022. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4741–4753.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902.

A Baseline Methods Details

This appendix provides comprehensive descriptions of the baseline methods compared in our experimental evaluation. All implementations strictly followed the original authors' specifications.

A.0.1 Compared Baseline Methods

In this study, we assess our proposed method, RaDAR, by conducting a comparative analysis against several baseline approaches to ensure a thorough evaluation. The specifics of these baseline methods are detailed below.

- **BiasMF**(Koren et al., 2009): This is a matrix factorization technique designed to improve personalized recommendations by integrating bias vectors for both users and items, thereby capturing individual user preferences more effectively.
- **NCF**(He et al., 2017): This approach employs a neural network architecture that substitutes the conventional dot-product operation in matrix factorization with multi-layered neural networks. This modification enables the model to learn intricate user-item interactions, thereby enhancing the quality of recommendations. For the purpose of our comparison, we implement the NeuMF variant of NCF.
- **AutoR**(Sedhain et al., 2015): This approach enhances the representation of users and items through a three-layer autoencoder trained with the objective of reconstructing interaction data.
- **GCMC**(Berg et al., 2017): This method utilizes graph convolutional networks (GCNs) for the task of completing interaction matrices.
- **PinSage**(Ying et al., 2018): This technique employs a graph convolutional framework augmented with random sampling to enhance performance in collaborative filtering tasks.
- **NGCF**(Wang et al., 2019): This model implements a multi-layer graph convolutional network that facilitates the propagation of information throughout the user-item interaction graph while learning latent representations for both users and items.
- **STGCN**(Zhang et al., 2019): This approach combines graph convolutional encoders with graph autoencoders to bolster the model's resilience against issues such as sparsity and cold-start scenarios in collaborative filtering applications.
- **LightGCN**(He et al., 2020): This model capitalizes on neighborhood information in the user-item interaction graph by employing a layer-wise propagation method that relies solely on linear transformations and element-wise summation.
- **GCCF**(Chen et al., 2020): This method presents a novel framework for collaborative filtering recommender systems by re-examining the application of graph convolutional networks. It addresses the over-smoothing issue by discarding non-linear activations and incorporating a residual network architecture.
- **HCCF**(Xia et al., 2022a): The authors introduce a self-supervised recommendation framework that adeptly captures both local and global collaborative interactions through the deployment of a hypergraph neural network augmented by a cross-view contrastive learning mechanism.
- **SHT**(Xia et al., 2022b): This approach synergistically combines hypergraph neural networks with transformers under a self-supervised learning paradigm, focusing on data augmentation to effectively reduce noise in user-item interaction data within recommendation systems.
- **SLRec**(Yao et al., 2021): The proposed model employs contrastive learning among node features as regularization techniques, thereby enhancing the efficacy of state-of-the-art collaborative filtering recommender systems.
- **SGL**(Wu et al., 2021): This model enhances LightGCN by integrating self-supervised contrastive learning, utilizing data augmentation strategies such as random walk and node/edge dropout to perturb graph structures.
- **NCL**(Lin et al., 2022): The neighborhood-enriched contrastive learning (NCL) approach enhances graph-based collaborative filtering by integrating potential neighbors into the formation of contrastive pairs. NCL delineates both structural and semantic neighbors for users or items, which facilitates the establishment of a structure-contrastive objective as well as a prototype-contrastive objective.
- **DirectAU**(Wang et al., 2022a): This novel tech-

nique presents a new learning objective specifically designed for collaborative filtering methodologies. It evaluates representation quality through alignment and uniformity on the hypersphere, thereby directly optimizing two essential properties to boost recommendation performance.

- **AdaGCL**(Jiang et al., 2023): This pioneering framework introduces an adaptive graph contrastive learning (AdaGCL) paradigm for collaborative filtering approaches. It utilizes two trainable view generators to produce contrastive views, enabling an adaptive mechanism for generating views tailored for contrastive learning within the collaborative filtering context.

B Mathematical Details

B.1 Embedding Propagation Details

The normalized adjacency matrix is computed as:

$$\tilde{\boldsymbol{A}} = \boldsymbol{D}_u^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}_v^{-\frac{1}{2}} \tag{9}$$

where $\boldsymbol{D}_u$ and $\boldsymbol{D}_v$ are diagonal degree matrices for users and items.

At the l-th layer, the embeddings are updated through:

$$\begin{aligned} \mathbf{E}_l^{(u)} &= \tilde{\mathbf{A}} \mathbf{E}_{l-1}^{(v)} + \mathbf{E}_{l-1}^{(u)} \\ \mathbf{E}_l^{(v)} &= \tilde{\mathbf{A}}^\top \mathbf{E}_{l-1}^{(u)} + \mathbf{E}_{l-1}^{(v)} \end{aligned} \tag{10}$$

The final embeddings are computed as:

$$\mathbf{E}^{(u)} = \sum_{l=0}^{L} \mathbf{E}_l^{(u)}, \quad \mathbf{E}^{(v)} = \sum_{l=0}^{L} \mathbf{E}_l^{(v)} \tag{11}$$

The preference score is calculated as:

$$\hat{y}_{i,j} = (e_i^{(u)})^\top e_j^{(v)} \tag{12}$$

B.2 Variational Graph Auto-Encoder Details

In this section, we provide the detailed mathematical formulations of the VGAE framework used in our view generation approach. The KL-divergence regularization term for the latent distributions is defined as:

$$\mathcal{L}_{\text{kl}} = -\frac{1}{2} \sum_{d=1}^{D} (1 + 2\log(\mathbf{x}_{\text{std}}) - \mathbf{x}_{\text{mean}}^2 - \mathbf{x}_{\text{std}}^2) \tag{13}$$

For graph structure reconstruction, we employ a discriminative loss $\mathcal{L}_{\text{kl}}$ that evaluates both positive and negative interactions:

$$\begin{aligned} \mathcal{L}_{\text{pos}} &= \text{BCE}(\sigma(f(\mathbf{x}_{\text{user}}[u] \odot \mathbf{x}_{\text{item}}[i])), \mathbf{1}) \\ &= -\log(\sigma(f(\mathbf{x}_{\text{user}}[u] \odot \mathbf{x}_{\text{item}}[i]))) \\ \mathcal{L}_{\text{neg}} &= \text{BCE}(\sigma(f(\mathbf{x}_{\text{user}}[u] \odot \mathbf{x}_{\text{item}}[j])), \mathbf{0}) \\ &= -\log(1 - \sigma(f(\mathbf{x}_{\text{user}}[u] \odot \mathbf{x}_{\text{item}}[j]))) \\ \mathcal{L}_{\text{dis}} &= \mathcal{L}_{\text{pos}} + \mathcal{L}_{\text{neg}} \end{aligned} \tag{14}$$

The Bayesian Personalized Ranking (BPR) loss is incorporated to enhance recommendation performance:

$$\mathcal{L}_{\text{bpr}} = \sum_{(u,i,j) \in O} -\log \sigma(\hat{y}_{ui} - \hat{y}_{uj}), \tag{15}$$

The total VGAE optimization objective combines these components with weight regularization:

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{kl}} + \mathcal{L}_{\text{dis}} + \mathcal{L}_{\text{bpr}}^{\text{gen}} + \lambda_2 \|\Theta\|_F^2, \tag{16}$$

B.3 Relation-Aware Graph Denoising Details

This section provides the mathematical details of our relation-aware graph denoising framework. The layer-wise edge masking with sparsity constraints is formulated as:

$$\begin{aligned} & A^l = A \odot M^l, \\ & \sum_{l=1}^{L} |M^l|_0 = \sum_{l=1}^{L} \sum_{(u,v) \in \epsilon} \mathbb{I}(m_{u,v}^l \neq 0) \end{aligned} \tag{17}$$

The denoising layer employs adaptive gating to preserve essential user-item relationships:

$$\begin{aligned} \mathbf{g} &= \sigma(\mathbf{W}_g[\mathbf{e}_i; \mathbf{e}_j] + \mathbf{b}) \\ \alpha_{i,j}^l &= f_{\text{att}}\left(\mathbf{G}(\mathbf{e}_i, \mathbf{e}_j) \oplus \mathbf{G}(\mathbf{e}_j, \mathbf{e}_i) \oplus [\mathbf{e}_i; \mathbf{e}_j]\right) \end{aligned} \tag{18}$$

The adaptive feature composition $\mathbf{G}[\cdot]$ is defined as:

$$\mathbf{G}(\mathbf{e}_i, \mathbf{e}_j) = \mathbf{g} \odot \tau(\mathbf{W}_{\text{embed}}[\mathbf{e}_i; \mathbf{a}_{r,i}]) + (1 - \mathbf{g}) \odot \mathbf{e}_i \tag{19}$$

The edge sampling employs a concrete distribution with hard sigmoid rectification:

$$\mathcal{L}_c = \sum_{l=1}^{L} \sum_{(u_i, v_j) \in \epsilon} \left(1 - \mathbb{P}\sigma(s_{i,j}^l \mid \theta^l)\right) \tag{20}$$

The final training objective combines concrete distribution regularization with recommendation loss:

$$\mathcal{L}_{\text{den}} = \mathcal{L}_c + \mathcal{L}_{\text{bpr}}^{\text{gen}} + \lambda_2 \|\Theta\|_F^2 \tag{21}$$

B.4 Detailed Diffusion Process Formulation

B.4.1 Forward Diffusion Process

Our diffusion process begins with the forward phase, where Gaussian noise is progressively added according to:

$$q(\boldsymbol{\chi}_t|\boldsymbol{\chi}_{t-1}) = \mathcal{N}(\boldsymbol{\chi}_t; \sqrt{1-\beta_t}\boldsymbol{\chi}_{t-1}, \beta_t \boldsymbol{I}) \quad (22)$$

with β_t controlling the noise scale at step t.

The intermediate state $\boldsymbol{\chi}_t$ can be efficiently computed directly from the initial state $\boldsymbol{\chi}_0$ through:

$$q(\boldsymbol{\chi}_t|\boldsymbol{\chi}_0) = \mathcal{N}(\boldsymbol{\chi}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{\chi}_0, (1-\bar{\alpha}_t)\boldsymbol{I}), \quad \bar{\alpha}_t = \prod_{t'=1}^{t}(1-\beta_{t'}) \quad (23)$$

This allows for the reparameterization:

$$\boldsymbol{\chi}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{\chi}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}) \quad (24)$$

B.4.2 Linear Noise Scheduler

To control the injection of noise in $\boldsymbol{\chi}_{1:T}$, we employ a linear noise scheduler that parameterizes $1-\bar{\alpha}_t$ using three hyperparameters:

$$1-\bar{\alpha}_t = s \cdot \left[\alpha_{low} + \frac{t-1}{T-1}(\alpha_{up} - \alpha_{low})\right], \quad t \in \{1, \cdots, T\} \quad (25)$$

Here, $s \in [0, 1]$ regulates the overall noise scale, while $\alpha_{low} < \alpha_{up} \in (0, 1)$ determine the lower and upper bounds for the injected noise.

B.4.3 Reverse Denoising Process

The reverse process aims to recover the original representations by progressively denoising $\boldsymbol{\chi}_t$ to reconstruct $\boldsymbol{\chi}_{t-1}$ through a neural network:

$$p_\theta(\boldsymbol{\chi}_{t-1}|\boldsymbol{\chi}_t) = \mathcal{N}(\boldsymbol{\chi}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{\chi}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{\chi}_t, t)) \quad (26)$$

where neural networks parameterized by θ generate the mean and covariance of the denoising distribution.

B.5 Asymmetric Contrastive Loss

The asymmetric contrastive learning loss function is defined as:

$$\mathcal{L}_A = -\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}}\frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)} \log\frac{\exp(p^\top u/\tau)}{\exp(p^\top u/\tau) + \sum_{v^-\in\mathcal{V}}\exp(v^\top v^-/\tau)}, \quad (27)$$

where $\mathcal{N}(v)$ represents the one-hop neighbors of node v, and τ controls the softmax temperature. The predictor output $p = g_\phi(v)$ transforms the identity representation into a prediction of its neighborhood context.