Revisiting Contrastive Learning in Collaborative Filtering via Parallel Graph Filters

Anonymous submission

Abstract

Graph Contrastive Learning (GCL) has recently emerged as a powerful paradigm for modeling user–item interactions and learning high-quality representations in recommender systems. While existing GCL-based methods benefit from data augmentation and sampling strategies, they often overlook the inherent limitations of the contrastive objectives: 1) Stacking multiple Graph Convolutional Network layers to capture high-order information often causes the over-smoothing phenomenon, where node representations become overly similar. 2) Structurally similar negative sample pairs may exhibit high cosine similarity, causing gradient saturation during representation optimization. To address the above challenges, we revisit matrix factorization in recommendation models and uncover its implicit connection to a parallel graph filter bank. This perspective reveals how overly aggressive low-pass or high-pass filtering distorts feature distributions, contributing to gradient saturation. Building on this insight, we propose Light Cosine Similarity Collaborative Filtering (LightCSCF), a margin-constrained method that improves gradient optimization in contrastive learning by focusing on structurally hard examples, alleviating both gradient saturation and boundary over-smoothing. Extensive experiments on three real-world datasets demonstrate that LightCSCF consistently outperforms state-of-the-art baselines in recommendation accuracy and robustness to data sparsity.

1 Introduction

In the contemporary digital landscape, recommender systems play a vital role in addressing information overload by guiding users toward personalized content. By analyzing user preferences, recommender systems enhance user experiences across a wide range of domains, including e-commerce, media streaming, and content discovery (Gao et al. 2023; Singh et al. 2021). The primary goal of recommender systems is to accurately model the complex and high-dimensional interactions between users and items, enabling effective personalization at scale.

Traditional collaborative filtering methods (Koren 2009; Rao et al. 2015) suffer from many challenges in recommender systems, including the absence of explicit collaborative signals and the cold-start problem (Lin et al. 2021; Liu et al. 2021). These challenges have motivated the exploration of graph-based models to learn representations. To better capture complex user-item interactions, Graph Convolutional Networks (GCN) have been introduced and have shown strong performance by modeling high-order connectivity patterns (He et al. 2020). However, GCN-based methods still face limitations, including data sparsity and long-tail distribution problems. To address these challenges, recent studies have turned to Contrastive Learning (CL) (Chuang et al. 2020; Khosla et al. 2020; Tian et al. 2020; Zhang and Wu 2024), a promising self-supervised learning paradigm. CL aims to construct a discriminative representation space by pulling positive pairs closer and pushing negative pairs apart. By deriving supervision signals directly from the data itself, CL enables the learning of generalizable representations, enhancing recommendation performance.

Building on this paradigm, a diverse array of recommendation methods has adopted CL to improve representation quality (Ma, Lian, and Song 2025; Wang et al. 2024). SGL (Wu et al. 2021) introduces a graph augmentation framework that perturbs the graph structure to generate informative self-supervised signals, while SimGCL (Yu et al. 2022), XSimGCL (Yu et al. 2023), and LightGCL (Cai et al. 2023) primarily explore augmentation by refining noise injection into the embeddings. Furthermore, methods such as CGCL (He et al. 2023), BIGCF (Zhang, Sang, and Zhang 2024) and SCCF (Wu et al. 2024) enhance the contrastive quality by further modeling user intention and multi-scale contrast. In parallel, a growing line of research begins to investigate the role of graph signal processing in CL (Liu et al. 2022a; Yu et al. 2023), in which contrastive views can be constructed to selectively emphasize low-pass or high-pass components via graph filter banks.

Despite continued advancements in CL methods, several fundamental obstacles remain hindering their representation quality (Yu et al. 2023): 1). Deepening GCN to capture high-order neighborhood relationships often results in over-smoothing, where node features become overly similar and lose their discriminative power; 2) The inclusion of hard negatives structurally similar to positives can cause gradient saturation by providing weak contrastive signals during training. These limitations motivate us to reconsider the problem from a spectral perspective. Inspired by GAME (Liu et al. 2022b), which emphasizes that the discrepancy in high-frequency components between two augmented graphs should exceed that in low-frequency components, we explore new directions for contrastive learning.

To address these challenges, we first conduct a theoretical analysis to further reveal the origins of over-smoothing in GCN. Specifically, as the number of GCN layers increases, repeated propagation through the normalized adjacency matrix progressively suppresses high-frequency components in node features. This leads to overly smooth representations, making them nearly indistinguishable across nodes and weakening their discriminative power.

We then investigate the origins of gradient saturation in CL from both geometric and theoretical perspectives. The gradient updates in CL are affected by cosine similarity, which causes gradient saturation for extreme embeddings. Meanwhile, we reveal that the low-rank projection of MF acts as a low-pass graph filter that promotes smoothness. This insight inspires us to unify embedding updates that integrate identity, low-pass, and high-pass operators to efficiently capture both global trends and local variations (as detailed in Section 3 and Section 4).

In this paper, we propose **Light** **C**osine **S**imilarity **C**ollaborative **F**iltering (**LightCSCF**), a novel method designed to effectively address margin gradient saturation. Motivated by the limitations of conventional cosine similarity, we propose a margin-constrained method that not only increases the similarity of positive pairs but also suppresses negative pairs that are structurally similar. The main contributions of this work are as follows:

- We theoretically derive the over-smoothing phenomenon in GCN and incorporate a geometric perspective to investigate the gradient saturation problem in CL.
- We pinpoint the limitations of cosine similarity and propose LightCSCF, a lightweight yet effective collaborative filtering method to efficiently mitigate graph over-smoothing and alleviate gradient saturation.
- We demonstrate through comprehensive experiments on three highly sparse datasets that LightCSCF significantly outperforms existing CL-based methods, achieving enhanced recommendation accuracy and improve robustness against data sparsity.

2 Preliminaries

Problem Definition. The fundamental goal of a recommender system is to model user preferences based on historical interactions, in order to accurately predict future interests. We represent the user-item interaction system as a bipartite graph $\mathcal{G} = (\mathcal{U} \cup \mathcal{I}, \mathcal{E})$, where $\mathcal{U} = \{u_1, \ldots, u_M\}$ and $\mathcal{I} = \{i_1, \ldots, i_N\}$ denote the sets of users and items, respectively, and $\mathcal{E}$ denotes the set of interactions between users and items. Let $\mathbf{A} \in \{0,1\}^{M \times M}$ be the adjacency matrix. We define $\mathbf{D}_u \in \mathbb{R}^{M \times M}$ and $\mathbf{D}_i \in \mathbb{R}^{N \times N}$ as the diagonal degree matrices for users and items, respectively.

Collaborative Filtering. Matrix Factorization (MF) (Lee and Seung 1999) serves as a foundational technique in collaborative filtering, where the goal is to approximate the user-item interaction matrix $\mathbf{R} \in \mathbb{R}^{M \times N}$ by learning low-dimensional latent embeddings. Specifically, MF defines this approximation as the factorization $\mathbf{R} \approx \mathbf{E}_u \mathbf{E}_i^\top$, where $\mathbf{E}_u \in \mathbb{R}^{M \times d}$ and $\mathbf{E}_i \in \mathbb{R}^{N \times d}$ are the user and item embedding matrices, respectively, and the predicted interaction score is the inner product $\hat{r}_{ui} = \langle \mathbf{e}_u, \mathbf{e}_i \rangle$. In contrast to learning a static set of embeddings, a GCN (Kipf and Welling 2017) layer refines node representations iteratively by aggregating information from their neighbors. This process begins with an initial learnable embedding matrix $\mathbf{E}^{(0)}$ for all users and items. At each layer l, the network updates these representations to produce a new matrix $\mathbf{E}^{(l)} \in \mathbb{R}^{(M+N) \times d}$.

Contrastive Learning. Contrastive Learning (Chen et al. 2020) (CL) is a self-supervised learning paradigm that aims to learn representations by pulling positive pairs closer and pushing negative pairs apart in the embedding space. Formally, given an anchor embedding $\mathbf{e}_u$, a positive embedding $\mathbf{e}_i$, and a set of negative embeddings $\{\mathbf{e}_j\}$, the contrastive loss encourages $\mathbf{e}_u$ to be more similar to $\mathbf{e}_i$ than to any negative $\mathbf{e}_j$. A commonly used form is the InfoNCE loss (Oord, Li, and Vinyals 2018):

$$\mathcal{L}_{\text{cl}} = -\log \frac{\exp(\text{sim}(\mathbf{e}_u, \mathbf{e}_i)/\tau)}{\exp(\text{sim}(\mathbf{e}_u, \mathbf{e}_i)/\tau) + \sum_j \exp(\text{sim}(\mathbf{e}_u, \mathbf{e}_j)/\tau)}, \tag{1}$$

where τ is the temperature coefficient controlling the concentration level, $\text{sim}(\mathbf{e}_u, \mathbf{e}_i)$ represents cosine similarity.

3 Investigation of GCN and CL

In this section, we primarily analyze the over-smoothing in GCN and the gradient saturation problem in CL.

3.1 Over-smoothing in GCN

GCN iteratively propagates information through the normalized adjacency matrix $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$. Under the simplification of removing non-linearities and setting all weight matrices to identity, this process reduces to the propagation rule $\mathbf{E}^{(K)} = \tilde{\mathbf{A}}^K \mathbf{E}^{(0)}$ (He et al. 2020), where repeated applications of $\tilde{\mathbf{A}}$ progressively smooth the node embeddings.

From a spectral perspective, the eigenvalues $\{\lambda_i\}$ of $\tilde{\mathbf{A}}$ lie within $[-1, 1]$. For a connected and non-bipartite graph, the largest eigenvalue $\lambda_1 = 1$ is simple, ensuring that $\tilde{\mathbf{A}}^K$ converges to a rank-one matrix as $K \to \infty$, with each row proportional to the principal eigenvector (Li, Han, and Wu 2018). This implies that high-frequency components associated with smaller eigenvalues $|\lambda_i| < 1$ diminish exponentially, causing node embeddings to asymptotically align along the same global direction. Consequently, the ability of the model to distinguish nodes degenerates with depth, which limits its effectiveness on tasks requiring local structural information. This trade-off between model depth and local discriminability in GCNs necessitates the design of a parallel graph filter bank to mitigate over-smoothing effects and preserve the diversity of node representations.

3.2 Gradient Saturation in CL

CL relies on gradient signals, which are derived from similarity loss functions, to effectively shape the embedding space (Khosla et al. 2020). A commonly used choice is the cosine similarity, which facilitates the learning of expressive representations by encouraging closeness between positive pairs and enforcing separation between negative pairs, including structurally similar ones.

Definition 1 *Given two embedding vectors $\mathbf{z}_u$ and $\mathbf{z}_i$ that are projected onto the unit n-sphere via ℓ_2-normalization, their cosine similarity is defined as:*

$$\cos(\boldsymbol{z}_u, \boldsymbol{z}_i) = \frac{\boldsymbol{z}_u^\top \boldsymbol{z}_i}{\|\boldsymbol{z}_u\| \cdot \|\boldsymbol{z}_i\|} = \boldsymbol{z}_u^\top \boldsymbol{z}_i. \tag{2}$$

Definition 2 *Given two unit vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$ with angle θ between them, a circular cone in $\mathbb{R}^d$ with axis along $\boldsymbol{a}$ and aperture angle θ is defined as:*

$$\mathcal{K}_\theta(\boldsymbol{a}) = \left\{ \boldsymbol{x} \in \mathbb{R}^d \setminus \{\mathbf{0}\} \;\middle|\; \frac{\boldsymbol{a}^\top \boldsymbol{x}}{\|\boldsymbol{x}\|} = \cos(\boldsymbol{a}, \boldsymbol{b}) \right\}. \tag{3}$$

Intersecting this cone with the unit $(d{-}1)$-sphere $\mathbb{S}^{d-1} = \{\boldsymbol{x} \in \mathbb{R}^d \mid \|\boldsymbol{x}\| = 1\}$ yields:

$$\mathcal{L}_\theta(\boldsymbol{a}) = \mathcal{K}_\theta(\boldsymbol{a}) \cap \mathbb{S}^{d-1} = \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \|\boldsymbol{x}\| = 1,\ \boldsymbol{a}^\top \boldsymbol{x} = \cos(\boldsymbol{a}, \boldsymbol{b}) \right\}. \tag{4}$$

This set $\mathcal{L}_\theta(\mathbf{a})$ consists of all unit vectors that share the same cosine similarity with $\mathbf{a}$ as $\mathbf{b}$ does. Geometrically, it forms the intersection between a cone centered at $\mathbf{a}$ and the unit n-sphere. Specifically, treating $\mathbf{a}$ as the anchor and $\mathbf{b}$ as the positive sample, any vector $\mathbf{c} \in \mathcal{L}_\theta(\mathbf{a})$ lies on the intersection of the unit n-sphere and the cone centered at $\mathbf{a}$ with aperture angle $\theta = \arccos(\mathbf{a}^\top \mathbf{b})$. By construction, $\mathbf{c}$ satisfies $\mathbf{a}^\top \mathbf{c} = \mathbf{a}^\top \mathbf{b}$ and thus shares the same cosine similarity with $\mathbf{a}$ as the positive sample $\mathbf{b}$. This makes $\mathbf{c}$ indistinguishable from $\mathbf{b}$ despite representing a structurally negative sample. As the angle θ decreases, the cone narrows and $\mathcal{L}_\theta(\mathbf{a})$ collapses toward the direction of $\mathbf{a}$, causing the cosine similarity $\mathbf{a}^\top \mathbf{b}$ to approach 1 and placing the anchor-positive pair in the saturation region of the loss function. This saturation effect weakens the optimization signal, making the model insensitive to structurally negatives residing in $\mathcal{L}_\theta(\mathbf{a})$, limiting the discriminative ability of learned representations.

From theoretical analysis, the optimization signal in CL can be defined as $\mathrm{sim}(\boldsymbol{e}_u, \boldsymbol{e}_i) = e^{\cos(\boldsymbol{e}_u, \boldsymbol{e}_i)}$, which is influenced by the derivative of the cosine similarity between the embedding vectors $\mathbf{e}_u$ and $\mathbf{e}_i$. When the angle $\theta_{ui} \in [0, \pi]$ between the two embeddings approaches the endpoints, the derivative vanishes and the gradient information becomes negligible:

$$\text{At } \theta_{ui} = 0 \text{ or } \pi, \frac{\partial \mathrm{sim}(\boldsymbol{e}_u, \boldsymbol{e}_i)}{\partial \theta_{ui}} = \frac{\partial e^{\cos(\theta_{ui})}}{\partial \theta_{ui}} = 0. \tag{5}$$

As a result, when two embeddings are either nearly identical or diametrically opposed, the learning signal degenerates. This saturation effect restricts the model's ability to make meaningful adjustments, limiting the refinement of representation learning in later stages of training.

4 Methodology

In this section, we analyze the gradient update process in CL through the lens of MF, demonstrating that this process can be interpreted as a parallel graph filter bank. Based on this insight, we further propose a margin-constrained method.

4.1 CL with MF as Parallel Graph Filter

The update process in MF implicitly acts as a graph convolution by smoothing node embeddings over the interaction graph. To clarify this connection, we analyze the mathematical structure of the low-rank projection operator and its effect on embedding smoothness.

Definition 3 *Given an embedding matrix $\mathbf{P} \in \mathbb{R}^{n \times d}$, the matrix $\mathbf{M} = \mathbf{P}\mathbf{P}^\top \in \mathbb{R}^{n \times n}$ is called a low-rank projection operator. It maps any input vector $x \in \mathbb{R}^{n \times d}$ to its projection $x' = \mathbf{M}x = \mathbf{P}(\mathbf{P}^\top x)$, which lies in the column space of $\mathbf{P}$.*

Definition 4 *The spectral norm $\|\cdot\|_2$ satisfies the sub-multiplicative property: for any compatible matrices $\mathbf{A}$ and $\mathbf{B}$, $\|\mathbf{AB}\|_2 \le \|\mathbf{A}\|_2 \cdot \|\mathbf{B}\|_2$.*

The projection matrix $\mathbf{M} = \mathbf{P}\mathbf{P}^\top$ maps input signals onto a low-dimensional subspace spanned by the learned embeddings. This operation suppresses high-frequency components in the original signal, acting as a low-pass filter. In the context of graph signal processing, such filtering promotes smoothness by reducing variations across connected nodes. To quantify this effect, we examine how the smoothness of the transformed signal Mx relates to the original signal.

Theorem 1 *Given a graph Laplacian matrix $\mathbf{L}$, the smoothness of a signal $\boldsymbol{x}$ is measured by $S(\boldsymbol{x}) = \boldsymbol{x}^\top \mathbf{L} \boldsymbol{x}$.*

proof 1 *Let $S(\boldsymbol{x}) = \boldsymbol{x}^\top \mathbf{L} \boldsymbol{x} = \boldsymbol{x}^\top (\mathbf{D} - \mathbf{A}) \boldsymbol{x} = \boldsymbol{x}^\top \mathbf{D} \boldsymbol{x} - \boldsymbol{x}^\top \mathbf{A} \boldsymbol{x}$, where $\mathbf{D}_{ii} = \sum_{j=1}^N \mathbf{A}_{ij}$. This expands as:*

$$\begin{aligned} S(\boldsymbol{x}) &= \sum_{i=1}^N \left(\sum_{j=1}^N \mathbf{A}_{ij} \right) x_i^2 - \sum_{i=1}^N \sum_{j=1}^N \mathbf{A}_{ij} x_i x_j \\ &= \sum_{i=1}^N \sum_{j=1}^N \mathbf{A}_{ij} (x_i^2 - x_i x_j). \end{aligned} \tag{2}$$

By symmetry, the quadratic form can be rewritten as:

$$S(\boldsymbol{x}) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \mathbf{A}_{ij} (x_i - x_j)^2. \tag{3}$$

Here, $(x_i - x_j)^2$ represents the squared difference between signal values at nodes i and j, and $\mathbf{A}_{ij}$ is the corresponding edge weight.

Theorem 2 *Let $\mathbf{M}$ be a linear projection operator and $\mathbf{L}$ be the graph Laplacian. For any signal $\boldsymbol{x}$, the smoothness of the transformed signal $\mathbf{M}\boldsymbol{x}$ is upper-bounded by*

$$S(\mathbf{M}\boldsymbol{x}) \le \|\mathbf{M}\|_2^2 \cdot \|\mathbf{L}\|_2 \cdot \|\boldsymbol{x}\|_2^2.$$

proof 2 *The smoothness of the signal $\mathbf{M}\boldsymbol{x}$ is defined as:*

$$S(\mathbf{M}\boldsymbol{x}) = (\mathbf{M}\boldsymbol{x})^\top \mathbf{L} (\mathbf{M}\boldsymbol{x}) = \boldsymbol{x}^\top \mathbf{M}^\top \mathbf{L} \mathbf{M} \boldsymbol{x}.$$

By applying the sub-multiplicative property of the spectral norm, we have:

$$\boldsymbol{x}^\top \mathbf{M}^\top \mathbf{L} \mathbf{M} \boldsymbol{x} \le \|\mathbf{M}^\top \mathbf{L} \mathbf{M}\|_2 \cdot \|\boldsymbol{x}\|_2^2.$$

Since $\|\mathbf{M}^\top \mathbf{L} \mathbf{M}\|_2 \le \|\mathbf{M}\|_2^2 \cdot \|\mathbf{L}\|_2$, we obtain the result:

$$S(\mathbf{M}\boldsymbol{x}) \le \|\mathbf{M}\|_2^2 \cdot \|\mathbf{L}\|_2 \cdot \|\boldsymbol{x}\|_2^2.$$

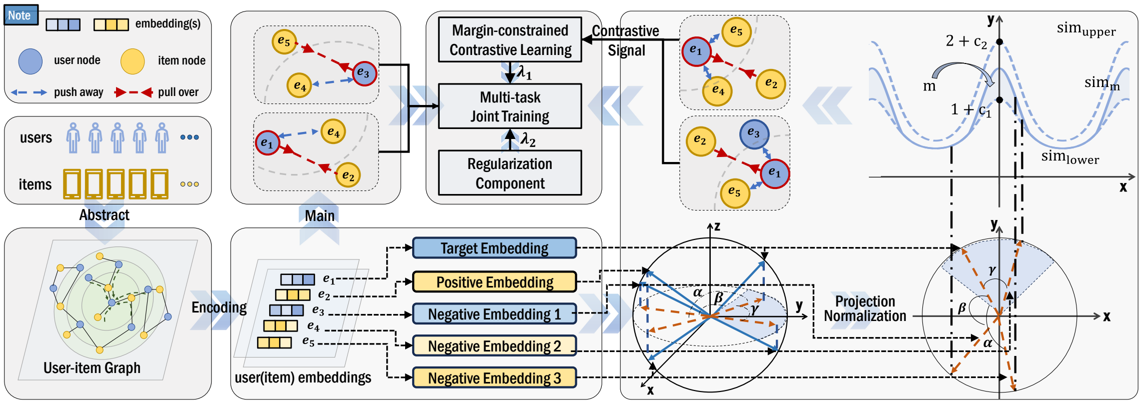


Figure 1: The LightCSCF method encodes user-item interactions via a bipartite graph, projects and normalizes the embeddings, then applies margin-constrained CL with multi-negative sampling for joint optimization of recommendation and representation.

Theorem 2 provides an upper bound on the smoothness of the transformed signal, showing that the projection operator $\mathbf{M}$ acts as a low-pass filter. Leveraging this property, we aim to guide the embedding $\mathbf{E}$ toward representations with desirable frequency characteristics. A natural method (Lee and Seung 2000) is to update embeddings by reinforcing low-frequency components via $\mathbf{M}$ and suppressing high-frequency components via the complementary high-pass operator $\mathbf{N} = \mathbf{I} - \mathbf{M}$. This leads to the embedding update rule:

$$\mathbf{E}^{(t+1)} = \mathbf{E}^{(t)} + \mu\mathbf{M}\mathbf{E}^{(t)} - \nu\mathbf{N}\mathbf{E}^{(t)}, \tag{6}$$

where μ and ν are coefficients. Unlike sequential message passing, this formulation applies the three filters in parallel, resembling a residual filtering mechanism that simultaneously captures smooth trends and discriminative variations.

We formalize the CL process through a parallel graph filter bank composed of three components: the identity operator $\mathbf{I}$, a low-pass filter $\mathbf{M}$ that promotes smoothness, and a high-pass filter $\mathbf{N}$ that captures high-frequency residuals. This unified design allows the contrastive gradient to be decomposed into interpretable signals, where the attractive gradient satisfies $-\eta\nabla_{\mathbf{E}}\mathcal{L}_{\mathrm{pull}} = \mu\mathbf{M}\mathbf{E}^{(t)}$ and the repulsive gradient satisfies $-\eta\nabla_{\mathbf{E}}\mathcal{L}_{\mathrm{push}} = -\nu\mathbf{N}\mathbf{E}^{(t)}$. Consequently, the embedding update follows $\mathbf{E}^{(t+1)} = (\mathbf{I}+\mu\mathbf{M}-\nu\mathbf{N})\mathbf{E}^{(t)}$, blending identity, low-pass, and high-pass operations within a unified filter bank framework. Furthermore, the repulsive signal from negative samples implicitly enforces low-rank constraints on the similarity matrix, acting mainly in the spectral domain. This connection underscores the importance of selecting structurally meaningful negative samples: while low-pass filters promote semantic smoothness, high-pass filters accentuate structural differences, thus enabling better discrimination between topologically similar yet semantically distinct nodes.

4.2 LightCSCF

To investigate how cosine similarity influences the CL loss in recommender systems, we analyze the loss function adopted by CL-based methods. Despite the growing interest in contrastive loss functions, we observe that few studies explicitly address the inherent limitations of the standard cosine similarity (Wu et al. 2021; Lin et al. 2022; Yu et al. 2022; Cai et al. 2023; He et al. 2023; Yang et al. 2023; Zhang, Sang, and Zhang 2024; Zhang et al. 2024c, 2025; Zhang and Zhang 2025). Among the existing efforts, VGCL (Yang et al. 2023) introduces a probabilistic framework which leverages the probability of two users (or items) being assigned to the same prototype to assist the cosine similarity in making similarity judgments, without altering the similarity function directly. Additionally, SCCF (Wu et al. 2024) enhances CL by modifying the similarity function $\mathrm{sim}(\cdot)$, defined as follows:

$$\mathrm{sim}(\boldsymbol{e}_u, \boldsymbol{e}_i) = \exp\left(\frac{\boldsymbol{e}_u^\top \boldsymbol{e}_i}{\tau\|\boldsymbol{e}_u\|_2\|\boldsymbol{e}_i\|_2}\right) + \exp\left(\frac{1}{\tau}\left(\frac{\boldsymbol{e}_u^\top \boldsymbol{e}_i}{\|\boldsymbol{e}_u\|_2\|\boldsymbol{e}_i\|_2}\right)^2\right), \tag{7}$$

where $\mathbf{e}_u$ and $\mathbf{e}_i$ denote the embedding vectors of nodes u and i. However, our analysis reveals a drawback: for hard negatives where $\mathbf{e}_u^\top\mathbf{e}_i \approx -1$, the squared term becomes positive, which potentially signals a false similarity.

Recognizing that hard negatives can generate misleading gradient signals and that similarity values in CL often saturate near their extremes, we reformulate the similarity function from a spectral perspective. Our similarity function consists of two exponential terms: the first corresponds to the conventional low-pass filtered cosine similarity, while the second introduces a margin-constrained rectification that activates when the similarity exceeds a threshold m, emphasizing high-frequency differences, thus aligning with the structure of the update operator $\mathbf{H} = \mathbf{I} + \mu\mathbf{M} - \nu\mathbf{N}$.

To better understand and control the gradient behavior introduced by the margin-constrained high-pass term, we analyze the theoretical bounds of our similarity formulation. First, the high-pass filter is designed to be sensitive to high-frequency signals and to amplify alignment signals between positive pairs. We therefore estimate the lower bound:

$$\mathrm{sim}_{\mathrm{lower}} = \exp\left(\frac{\cos(\boldsymbol{e}_u, \boldsymbol{e}_i)}{\tau}\right) + c_1, \tag{8}$$

where c_1 is a positive constant that ensures a non-vanishing contribution from the low-pass term while providing a consistent gain across positive samples. Empirically, we choose $c = 1$, consistent with the design of the similarity function used in SCCF. Next, we consider an upper bound beyond which the high-pass filter may dominate overall similarity, contradicting our design goal of balanced contributions from both frequency bands. We define the upper bound as

$$\text{sim}_{\text{upper}} = \gamma \cdot \exp\left(\frac{\cos(\boldsymbol{e}_u, \boldsymbol{e}_i)}{\tau}\right) + c_2, \tag{9}$$

where c_2 is a positive constant and $\gamma \leq 2$ ensures that the low-frequency structure remains the primary contributor to smoothness, while the high-frequency component acts as a targeted rectifier. Then we define the final margin-constrained similarity as

$$\text{sim}_{\text{m}} = \exp\left(\frac{\cos(\boldsymbol{e}_u, \boldsymbol{e}_i)}{\tau}\right) + G(\boldsymbol{e}_u, \boldsymbol{e}_i; \tau), \tag{10}$$

where $G(\boldsymbol{e}_u, \boldsymbol{e}_i; \tau)$ is the high-pass filter that satisfies

$$\text{sim}_{\text{lower}} \leq \text{sim}_{\text{m}} \leq \text{sim}_{\text{upper}}. \tag{11}$$

We instantiate G using the constraint function with a margin parameter m as $\exp\left(\max(\cos(\mathbf{e}_u^\top \mathbf{e}_i) - m, 0)/\tau\right)$ that introduces piecewise non-linearity while preserving gradient flow. The operational process of our method is shown in Figure 1. Based on the bounded formulation, we design a loss function that optimizes the margin-constrained similarity:

$$\mathcal{L}_{\text{LightCSCF}} = -\sum_{i \in \mathcal{S}_u} \log \text{sim}_{\text{m}}(\boldsymbol{e}_u, \boldsymbol{e}_i) + \log\left(\sum_{j \in \mathcal{B}_u} \text{sim}_{\text{m}}(\boldsymbol{e}_u, \boldsymbol{e}_j)\right). \tag{12}$$

Here, e_u and e_i denote the embedding vectors for user u and item i, respectively. In the loss formulation, $\mathcal{S}_u$ refers to the set of positive items that user u has interacted with, while $\mathcal{B}_u$ denotes the set of other users and items in the same training batch, which serve as implicit negative samples. When the cosine similarity between the anchor and a positive sample exceeds the margin m, sim_{m} increases the contribution of this pair through the second term, effectively pulling the positive sample closer. On the other hand, if the similarity between the anchor and a sample from $\mathcal{B}_u$ does not exceed the margin, the second term degenerates into a constant, reducing the overall similarity and pushing apart structurally similar but undesired negative samples.

Our proposed margin-constrained method offers several key advantages: (1) the margin constraint selectively emphasizes highly similar pairs, enhancing the model's ability to separate positives from negatives in a margin-constrained metric learning manner; (2) to address gradient saturation, an additional term provides refinement signals when similarities exceed the margin, alleviating vanishing gradients for aligned pairs; (3) the piecewise-smooth activation function helps mitigate over-smoothing by preserving informative gradients and introducing non-linearity at zero.

4.3 Multi-task Joint Training

Although $\mathcal{L}_{\text{LightCSCF}}$ alleviates the problems of gradient saturation and over-smoothing, its implicit supervision may be insufficient for optimizing on the core CF task. To address this, we adopt a multi-task learning framework that jointly optimizes the model using both the BPR loss (Rendle et al. 2009) and the LightCSCF loss. The overall training objective is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{main}} + \lambda_1 \mathcal{L}_{\text{LightCSCF}} + \lambda_2 \mathcal{L}_{\text{reg}}, \tag{13}$$

where λ_1 balances the impact of the LightCSCF loss, λ_2 denotes the regularization strength, and $\mathcal{L}_{\text{reg}}$ represents the regularization loss.

Dataset	#Users	#Items	#Interactions	Density
Amazon-book	52643	91599	2984108	0.06%
Douban-book	13024	22347	792062	0.27%
Tmall	47939	41390	2619389	0.13%

Table 1: Statistics of the datasets.

4.4 Theoretical Analysis

Let $s = \cos(\boldsymbol{e}_u, \boldsymbol{e}_i)$ denote the cosine similarity between node embeddings filtered by the parallel graph filter bank $\mathbf{H} = \mathbf{I} + \mu\mathbf{M} - \nu\mathbf{N}$, which serves as the basis for defining our similarity function.

The similarity function is defined as

$$\text{sim}_{\text{m}}(s) = \exp\left(\frac{s}{\tau}\right) + \exp\left(\frac{\max(s - m, 0)}{\tau}\right). \tag{14}$$

The first and second derivatives with respect to s are

$$\text{sim}_{\text{m}}^{(1,2)}(s) = \begin{cases} \frac{1}{\tau}\exp\left(\frac{s}{\tau}\right), & \text{if } s \leq m, \\ \frac{1}{\tau}\left[\exp\left(\frac{s}{\tau}\right) + \exp\left(\frac{s-m}{\tau}\right)\right], & \text{if } s > m, \end{cases} \tag{15}$$

This piecewise smooth function causes the gradient magnitude $\text{sim}'_{\text{m}}(s)$ to increase more rapidly when $s > m$, imposing stronger penalties on hard negative pairs with high similarity. Moreover, by amplifying the gradients for high-similarity negative pairs, the margin-based similarity function implicitly preserves discriminative features across similar nodes, mitigating the over-smoothing effect that often occurs in deep GCN-based CL. Simultaneously, the increased curvature $\text{sim}''_{\text{m}}(s)$ in this regime may improve optimization stability and convergence, while also enhancing sensitivity to subtle differences among highly similar embeddings by providing more informative gradient signals.

5 Experiments

In this section, we conduct extensive experiments on three real-world datasets and compare LightCSCF with several state-of-the-art methods to evaluate its effectiveness. The statistics of the datasets are summarized in Table 1.

5.1 Experimental Setting

Datasets and Evaluation Metrics. We utilize three widely used public benchmark datasets in our experiments: *Amazon-book* (Sang et al. 2025), *Douban-book* (Yu et al. 2022), and *Tmall* (Ren et al. 2023). And we randomly split each dataset into training and test sets with a ratio of 80% to 20%, respectively, and adopt two standard metrics: Recall@K and NDCG@K, where $K \in \{10, 20\}$.

Dataset	Amazon-book				Tmall				Douban-book			
Method	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 (UAI'09)	0.0170	0.0182	0.0308	0.0239	0.0312	0.0287	0.0547	0.0400	0.0849	0.1079	0.1292	0.1147
DirectAU (KDD'22)	0.0296	0.0297	0.0506	0.0401	0.0475	0.0443	0.0752	0.0576	0.1153	0.1527	0.1660	0.1568
LGCN++ (RS'24)	0.0241	0.0257	0.0425	0.0330	0.0493	0.0465	0.0796	0.0595	0.1254	0.1695	0.1790	0.1729
LGODE (CIKM'24)	0.0267	0.0282	0.0453	0.0356	0.0478	0.0450	0.0777	0.0579	0.0989	0.1242	0.1347	0.1272
Mult-VAE (WWW'18)	0.0224	0.0239	0.0407	0.0315	0.0467	0.0423	0.0740	0.0552	0.1156	0.1532	0.1670	0.1604
CVGA (TOIS'23)	0.0290	0.0302	0.0492	0.0379	0.0540	0.0517	0.0854	0.0648	0.1229	0.1670	0.1763	0.1699
DiffRec (SIGIR'23)	0.0310	0.0333	0.0514	0.0418	0.0485	0.0473	0.0792	0.0612	0.1102	0.1545	0.1619	0.1661
SGL (SIGIR'21)	0.0263	0.0281	0.0478	0.0379	0.0457	0.0434	0.0738	0.0556	0.1153	0.1558	0.1633	0.1585
SimGCL (SIGIR'22)	0.0313	0.0334	0.0515	0.0414	0.0559	0.0536	0.0884	0.0674	0.1230	0.1637	0.1772	0.1583
LightGCL (ICLR'23)	0.0303	0.0318	0.0506	0.0397	0.0531	0.0508	0.0833	0.0637	0.1069	0.1393	0.1570	0.1455
BIGCF (SIGIR'24)	0.0294	0.0320	0.0500	0.0398	0.0547	0.0524	0.0876	0.0664	0.1199	0.1642	0.1741	0.1682
MixRec (WWW'25)	0.0325	0.0351	0.0534	0.0427	0.0481	0.0458	0.0757	0.0577	0.1149	0.1531	0.1668	0.1578
LightCCF(SIGIR'25)	0.0340	0.0367	0.0569	0.0455	0.0592	0.0564	0.0930	0.0706	0.1321	0.1851	0.1863	0.1854
Improv.1%	5.00%	4.09%	4.39%	3.74%	4.39%	4.96%	4.52%	4.96%	2.04%	1.67%	1.93%	1.46%
SCCF (KDD'24)	0.0221	0.0248	0.0378	0.0307	0.0438	0.0415	0.0713	0.0533	0.1255	0.1666	0.1773	0.1696
RecDCL (WWW'24)	0.0311	0.0318	0.0525	0.0407	0.0527	0.0492	0.0853	0.0632	0.1151	0.1452	0.1664	0.1526
LightCSCF (Ours)	**0.0357***	**0.0382***	**0.0594***	**0.0472***	**0.0618***	**0.0592***	**0.0972***	**0.0741***	**0.1348***	**0.1882***	**0.1899***	**0.1881***
Improv.2%	14.79%	20.13%	13.14%	15.97%	17.27%	20.33%	13.95%	17.25%	17.12%	29.61%	14.12%	23.26%

Table 2: A comparison between LightCSCF and state-of-the-art baselines is shown, with best results in **bold** and second-best underlined. 'Improv.1%' and 'Improv.2%' indicate relative improvements over CL-based baseline and the RCL-based baseline. An asterisk (*) marks statistically significant improvements (t-test $p < 0.05$) over the second-best.

Baselines. The baseline models used for comparison can be categorized into four groups: (1) **BPR- and AU-based methods**: MF-BPR (Koren, Bell, and Volinsky 2009), as well as the representation learning process through alignment and uniformity losses: DirectAU (Wang et al. 2022). (2) **GCN-based methods**: LightGCN++ (Lee, Kim, and Shin 2024), LightGODE (Zhang et al. 2024b). (3) **Generative-based methods**: Mult-VAE (Liang et al. 2018), CVGA (Zhang et al. 2023), DiffRec (Wang et al. 2023). (4) **CL-based methods**: SGL (Wu et al. 2021), SimGCL (Yu et al. 2022), LightGCL (Cai et al. 2023), BIGCF (Zhang, Sang, and Zhang 2024), LightCCF (Zhang et al. 2025), MixRec (Zhang and Zhang 2025). (5) **RCL-based methods**: SCCF (Wu et al. 2024), RecDCL (Zhang et al. 2024a).

Implementation Details. To ensure a fair comparison, all experiments were conducted using a unified framework. Xavier initialization was applied to all models, and the embedding size was fixed at 64. An exception is made for RecDCL, which requires the embedding size to be set to 2048 due to its architectural design. For GCN-based methods, the number of layers is set to 3. All models are trained for up to 1000 epochs with an early stopping patience of 20. The temperature coefficient τ is set within $\{0.1, 0.2, 0.3, 0.5, 0.75, 1.0\}$ and the hyperparameter m is set within $\{0.1, 0.2, 0.3, ..., 1\}$. The loss weight α in LightCSCF is set within $\{0.1, 0.2, 0.3, 0.5, 1, 2.5, 5, 10\}$. The regularization weight β and the learning rate are fixed at 0.0001 and 0.001, respectively.

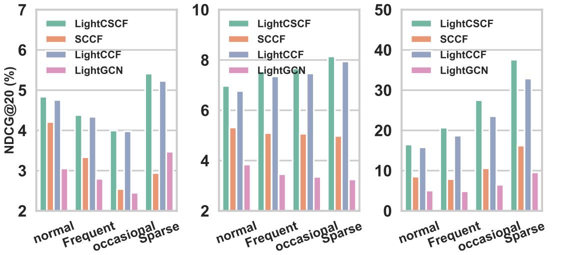
5.2 Performance Compare with Baselines

As shown in Table 2, LightCSCF consistently outperforms all baselines across three datasets. Specifically, it improves Recall@20 over the second-best model by 4.39%, 4.52%, and 1.93% on Amazon-Book, Tmall, and Douban-book, respectively. Compared with RCL-based methods, LightCSCF achieves even greater gains of 13.14%, 13.95%, and 14.12%. These results highlight the ability of LightCSCF to identify structurally similar negative samples.

LightCSCF also shows significant advantages over various CL-based models such as SimGCL, MixRec, and LightCCF, which depend on complex augmentations or neighborhood aggregation. These methods are still constrained by standard cosine similarity. In contrast, LightCSCF surpasses them without extra augmentation, demonstrating the effectiveness of its boundary-based cosine similarity method. Against advanced RCL models like SCCF and RecDCL, LightCSCF achieves substantial gains, including a 23.26% improvement in NDCG@20 on Douban-Book. Overall, LightCSCF adopts a margin-constrained method that improves the discrimination of structurally similar negatives while reducing reliance on graph convolutions.

5.3 In-depth Studies of LightCSCF

Sparsity Analysis. To further investigate the data sparsity issue, we conducted sparsity-based experiments on LightCSCF and competitive baselines by dividing the dataset into four user interaction frequency categories: frequent, normal, occasional, and sparse, then measuring performance via NDCG@20. As shown in Figure 2, LightCSCF consistently outperforms baselines across all sparsity levels thanks to its margin-constrained cosine similarity method. Notably, in the Amazon-Book dataset, the sparse subset achieves higher scores than the other categories, which can be explained by the prevalence of false-positive samples weakening the

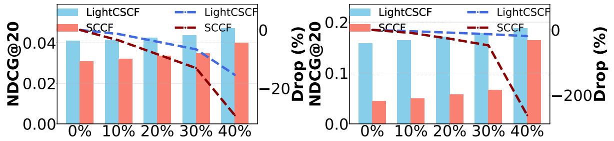


(a) Amazon-book (b) Tmall (c) Douban-book

Figure 2: Performance comparison of LightCSCF with other models under different sparsity levels on three datasets.

effect of $\mathcal{L}_{\text{BPR}}$. Removing $\mathcal{L}_{\text{BPR}}$ allows the influence of $\mathcal{L}_{\text{LightCSCF}}$ to stand out more clearly.

Robustness Analysis. As illustrated in Figure 3, the bar chart shows the changes of two models under different noise levels, while the line chart depicts the performance drop ratio relative to the performance without noise injection. We observe that NDCG@20 increases as more noise is introduced. This counterintuitive trend can be attributed to the mitigation of overfitting, which in turn enhances the generalization ability of the model. Furthermore, we find that LightCSCF exhibits minimal performance degradation across both datasets, highlighting the strong discriminative capacity of its loss function $\mathcal{L}_{\text{LightCSCF}}$, which effectively alleviates gradient saturation and over-smoothing.

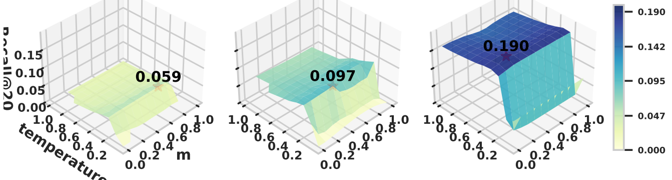


(a)Amazon-book (b)Douban-book

Figure 3: Robustness comparison of LightCSCF.

Hyperparameter Analysis. LightCSCF introduces only two hyperparameters, m and τ, keeping the overall number of hyperparameters lower than that of the SOTA CL-based methods. To better understand the influence of m, we conducted a grid search over selected values to identify the optimal setting for three benchmark datasets, which are presented in Figure 4. The best Recall@20 scores achieved were 0.0594, 0.0972, and 0.1898, respectively. As predicted by our theoretical analysis, the proposed margin-constrained cosine similarity method $\text{sim}_{\text{LightCSCF}}$ modifies the standard cosine similarity to some extent. While τ remains a key factor in CL loss, significantly impacting the uniformity of the learned embedding distribution (Zhang et al. 2024c).

Embedding Size Analysis. As shown in Figure 5, model performance improves as the embedding dimension increases, but tends to saturate after a certain point. This trend reflects the trade-off between representational capacity and efficiency. Beyond computational considerations, the effectiveness of lower-dimensional embeddings can be explained by classical MF methods, which are based on the assumption that a small number of leading singular components can sufficiently approximate the interaction matrix (Wu et al. 2024).



(a) Amazon-book (b) Tmall (c) Douban-book

Figure 4: Grid searches of impact on m and τ of LightCSCF on three datasets. Star represents optimal performance.

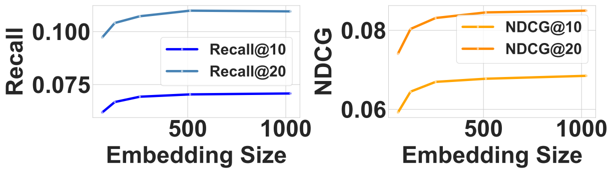
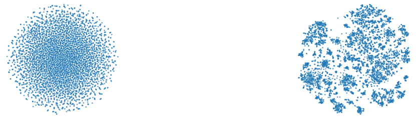


Figure 5: Performance of different embedding sizes (Tmall).

t-SNE Visualization Analysis. To evaluate the representation of LightCSCF, we randomly sampled 5,000 user and item embeddings, and projected them to a 2D surface using t-SNE in Figure 6. The resulting visualization reveals a reduced clustering effect among users and items, with user embeddings exhibiting a more uniform distribution. This suggests that the margin-constrained method effectively identifies hard negative samples with structural similarity, mitigating the formation of clusters.



(a) User embeddings (b) Item embeddings

Figure 6: t-SNE visualization of LightCSCF (Tmall).

6 Conclusion

In this paper, we theoretically show that deep graph convolution intensifies the over-smoothing problem. We analyze gradient saturation in CL from both geometric and theoretical perspectives, revealing its connection to MF. Based on these insights, we propose LightCSCF, a novel recommendation method that alleviates both over-smoothing and gradient saturation via a margin-constrained method. This approach suppresses hard negatives that are structurally similar to the anchor, encouraging clearer separation between positive and negative pairs in the embedding space. Extensive experiments on three real-world datasets confirm the performance and robustness of LightCSCF.

References

Cai, X.; Huang, C.; Xia, L.; and Ren, X. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PmLR.

Chuang, C.-Y.; Robinson, J.; Lin, Y.-C.; Torralba, A.; and Jegelka, S. 2020. Debiased contrastive learning. In *Advances in neural information processing systems*, volume 33, 8765–8775.

Gao, C.; Zheng, Y.; Li, N.; Li, Y.; Qin, Y.; Piao, J.; Quan, Y.; Chang, J.; Jin, D.; He, X.; et al. 2023. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems*, 1(1): 1–51.

He, W.; Sun, G.; Lu, J.; and Fang, X. S. 2023. Candidate-aware graph contrastive learning for recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, 1670–1679.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 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*, 639–648.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. In *Advances in neural information processing systems*, volume 33, 18661–18673.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Koren, Y. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 447–456.

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

Lee, D.; and Seung, H. S. 2000. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, volume 13.

Lee, D. D.; and Seung, H. S. 1999. Learning the parts of objects by non-negative matrix factorization. *nature*, 401(6755): 788–791.

Lee, G.; Kim, K.; and Shin, K. 2024. Revisiting LightGCN: Unexpected Inflexibility, Inconsistency, and A Remedy Towards Improved Recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems*, 957–962.

Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 3538–3545.

Liang, D.; Krishnan, R. G.; Hoffman, M. D.; and Jebara, T. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, 689–698.

Lin, X.; Wu, J.; Zhou, C.; Pan, S.; Cao, Y.; and Wang, B. 2021. Task-adaptive neural process for user cold-start recommendation. In *Proceedings of the web conference 2021*, 1306–1316.

Lin, Z.; Tian, C.; Hou, Y.; and Zhao, W. X. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM web conference 2022*, 2320–2329.

Liu, J.; Shi, C.; Yang, C.; Lu, Z.; and Yu, P. S. 2022a. A survey on heterogeneous information network based recommender systems: Concepts, methods, applications and resources. *AI Open*, 3: 40–57.

Liu, N.; Wang, X.; Bo, D.; Shi, C.; and Pei, J. 2022b. Revisiting graph contrastive learning from the perspective of graph spectrum. In *Advances in Neural Information Processing Systems*, volume 35, 2972–2983.

Liu, W.; Su, J.; Chen, C.; and Zheng, X. 2021. Leveraging distribution alignment via stein path for cross-domain cold-start recommendation. In *Advances in Neural Information Processing Systems*, volume 34, 19223–19234.

Ma, R.; Lian, Y.; and Song, C. 2025. Sub-Interest-Aware Representation Uniformity for Recommender System. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 12346–12354.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. arXiv:1807.03748.

Rao, N.; Yu, H.-F.; Ravikumar, P. K.; and Dhillon, I. S. 2015. Collaborative Filtering with Graph Information: Consistency and Scalable Methods. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Ren, X.; Xia, L.; Zhao, J.; Yin, D.; and Huang, C. 2023. Disentangled contrastive collaborative filtering. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, 1137–1146.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In Bilmes, J. A.; and Ng, A. Y., eds., *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, 452–461. AUAI Press.

Sang, L.; Zhang, Y.; Zhang, Y.; Li, H.; and Zhang, Y. 2025. Towards similar alignment and unique uniformity in collaborative filtering. *Expert Systems with Applications*, 259: 125346.

Singh, P. K.; Pramanik, P. K. D.; Dey, A. K.; and Choudhury, P. 2021. Recommender systems: an overview, research trends, and future directions. *International Journal of Business and Systems Research*, 15(1): 14–52.

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What makes for good views for contrastive learning? In *Advances in neural information processing systems*, volume 33, 6827–6839.

Wang, C.; Yu, Y.; Ma, W.; Zhang, M.; Chen, C.; Liu, Y.; and Ma, S. 2022. Towards representation alignment and uniformity in collaborative filtering. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 1816–1825.

Wang, K.; Zhu, Y.; Zang, T.; Wang, C.; and Jing, M. 2024. Enhanced hierarchical contrastive learning for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 9107–9115.

Wang, W.; Xu, Y.; Feng, F.; Lin, X.; He, X.; and Chua, T.-S. 2023. Diffusion recommender model. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, 832–841.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 726–735.

Wu, Y.; Zhang, L.; Mo, F.; Zhu, T.; Ma, W.; and Nie, J.-Y. 2024. Unifying graph convolution and contrastive learning in collaborative filtering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3425–3436.

Yang, Y.; Wu, Z.; Wu, L.; Zhang, K.; Hong, R.; Zhang, Z.; Zhou, J.; and Wang, M. 2023. Generative-contrastive graph learning for recommendation. In *Proceedings of the 46th international ACM SIGIR Conference on Research and Development in Information Retrieval*, 1117–1126.

Yu, J.; Xia, X.; Chen, T.; Cui, L.; Hung, N. Q. V.; and Yin, H. 2023. XSimGCL: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 36(2): 913–926.

Yu, J.; Yin, H.; Xia, X.; Chen, T.; Cui, L.; and Nguyen, Q. V. H. 2022. 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*, 1294–1303.

Zhang, D.; Geng, Y.; Gong, W.; Qi, Z.; Chen, Z.; Tang, X.; Shan, Y.; Dong, Y.; and Tang, J. 2024a. Recdcl: Dual contrastive learning for recommendation. In *Proceedings of the ACM Web Conference 2024*, 3655–3666.

Zhang, P.; and Wu, M. 2024. Multi-Label Supervised Contrastive Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16786–16793.

Zhang, W.; Yang, L.; Song, Z.; Zou, H. P.; Xu, K.; Fang, L.; and Yu, P. S. 2024b. Do we really need graph convolution during training? light post-training graph-ode for efficient recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 3248–3258.

Zhang, Y.; Sang, L.; and Zhang, Y. 2024. Exploring the individuality and collectivity of intents behind interactions for graph collaborative filtering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1253–1262.

Zhang, Y.; and Zhang, Y. 2025. MixRec: Individual and Collective Mixing Empowers Data Augmentation for Recommender Systems. In *Proceedings of the ACM on Web Conference 2025*, 2198–2208.

Zhang, Y.; Zhang, Y.; Sang, L.; and Sheng, V. S. 2024c. Simplify to the limit! embedding-less graph collaborative filtering for recommender systems. *ACM Transactions on Information Systems*, 43(1): 1–30.

Zhang, Y.; Zhang, Y.; Yan, D.; Deng, S.; and Yang, Y. 2023. Revisiting graph-based recommender systems from the perspective of variational auto-encoder. *ACM Transactions on Information Systems*, 41(3): 1–28.

Zhang, Y.; Zhang, Y.; Zhang, Y.; Sang, L.; and Yang, Y. 2025. Unveiling Contrastive Learning's Capability of Neighborhood Aggregation for Collaborative Filtering. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1985–1994.

Reproducibility Checklist

Instructions for Authors:

This document outlines key aspects for assessing reproducibility. Please provide your input by editing this `.tex` file directly.

For each question (that applies), replace the "Type your response here" text with your answer.

Example: If a question appears as

```
\question{Proofs of all novel claims
are included} {(yes/partial/no)}
Type your response here
```

you would change it to:

```
\question{Proofs of all novel claims
are included} {(yes/partial/no)}
yes
```

Please make sure to:

- Replace ONLY the "Type your response here" text and nothing else.
- Use one of the options listed for that question (e.g., **yes**, **no**, **partial**, or **NA**).
- **Not** modify any other part of the `\question` command or any other lines in this document.

You can `\input` this .tex file right before `\end{document}` of your main file or compile it as a stand-alone document. Check the instructions on your conference's website to see if you will be asked to provide this checklist with your paper or separately.

The questions start here

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) yes

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) yes

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) yes

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) yes

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) yes

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) yes

2.4. Proofs of all novel claims are included (yes/partial/no) yes

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) yes

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) yes

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) yes

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) yes

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) yes

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) yes

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) NA

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) NA

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) yes

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) yes

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) NA

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) yes

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) yes

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) yes

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) yes

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) yes

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) yes

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) yes

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) yes

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) yes

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) yes

4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) yes

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) yes

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) yes