Simple Yet Effective Heterogeneous Graph Contrastive Learning for Recommendation

Lei Sang, Haiyan Zhang, Yu Wang, and Yiwen Zhang

***Abstract*—Contrastive learning has emerged as a prominent approach in Heterogeneous Graph Neural Network (HGNN)-based recommender systems, exhibiting particular efficacy in addressing the sparsity challenge inherent in user-item interaction data. However, these heterogeneous contrastive learning recommendation models still face two main limitations: (1) Contrastive learning methods typically generate contrastive views through node dropping, edge perturbation, and random augmentation techniques, which not only have high complexity but may also disrupt the graph structure and introduce noise. (2) Most HGNN-based contrastive recommenders are susceptible to the phenomenon of over-smoothing, leading to indistinguishable representation problem. In this study, we introduce SimHGCL, a novel and efficient heterogeneous graph contrastive learning recommendation model. Our approach innovates by generating only two matrices derived from meta-paths on the user-item interaction graph, thereby eliminating the need for computationally expensive node dropping and edge perturbation operations. By leveraging diverse matrices based on meta-paths, we facilitate the creation of distinct contrastive views, thereby substantially streamlining the contrastive view generation process. Moreover, to address the challenge of indistinguishable representation resulting from over-smoothing in contrastive recommenders, we propose a learnable Laplacian matrix exponent. Comprehensive empirical evaluations conducted on four real-world datasets demonstrate that SimHGCL significantly outperforms state-of-the-art baselines in terms of standard normalized discounted cumulative gain and recall metrics, while concurrently achieving substantial reductions in computational time. To facilitate the reproducibility of our work, we have open-sourced our code at https://github.com/zhanghaiyan1018/SimHGCL.**

***Index Terms*—Consumer recommendation, contrastive learning, graph neural networks, heterogeneous graph representation.**

I. Introduction

Due to the exponential growth of information, online platforms are inundated with vast amounts of data, making it challenging to extract useful insights. Recommender systems [1], [2], [3], [4] have thus become essential for providing personalized services. Collaborative Filtering (CF) [5], [6], [7], a foundational algorithm in recommender systems, operates by exploiting user or item similarities and is widely applied across recommendation scenarios. However, traditional CF methods, relying mainly on matrix factorization, focus on direct one-hop connections in user-item interactions, limiting their ability to capture complex relational dynamics [8], [9], [10], [11]. To overcome these limitations, Graph Neural Networks (GNNs) have gained prominence in recommender systems [12], [13], [14], [15] for their ability to model higher-order interactions and aggregate collaborative signals effectively. Despite these advancements, many traditional and GNN-based models [16] still focus on homogeneous user-item bipartite graphs, neglecting the rich heterogeneous information prevalent in real-world scenarios [12]. This limitation exacerbates the sparsity problem, underscoring the need for more sophisticated approaches to leverage complex and diverse data structures.

Heterogeneous Information Networks (HINs) [12], [13], characterized by diverse node and edge types, provide a nuanced representation of complex systems and have become a research focus. Leveraging Heterogeneous Graph Neural Networks (HGNNs) for recommendation tasks on HINs has shown promise in addressing data sparsity [17]. Key to HINs are meta-paths, which capture intricate relationships and semantic associations among node types, aiding in uncovering community structures and higher-order relationships, as seen in social network analysis. Recent advancements in HGNN frameworks have significantly enhanced their capabilities. For instance, Heterogeneous Graph Attention Network (HAN) [18] employs node-level and semantic-level attention to extract complex topology and semantic information from HINs. Similarly, FD-HGL [19], leveraging a Heterogeneous Failure Dependency Graph (HFDG), attention mechanisms, and Relational Graph Convolutional Networks (RGCN) to achieve fine-grained failure diagnosis and root cause localization in microservice-based industrial clouds. The success of these HGNN-based methods across various scenarios highlights their potential in advancing recommender systems and other domains.

To address this need for enhanced representation learning in recommender systems, recent research has turned to contrastive learning [20], [21], [22], [23], [24], [25], a paradigm that has gained widespread adoption across diverse recommendation tasks owing to its formidable capabilities. Through the application of alignment and consistency

Received 30 August 2024; revised 22 November 2024, 6 January 2025, 13 February 2025, and 26 March 2025; accepted 11 May 2025. Date of publication 15 May 2025; date of current version 7 November 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62206002 and Grant 62272001; in part by the Anhui Provincial Natural Science Foundation under Grant 2208085QF195 and Grant 2308085MF22; in part by the Key Research Project of Philosophy and Social Science in Anhui Universities under Grant 2022AH050422; and in part by the Natural Science Foundation of Sichuan Province under Grant 2023NSFSC1420. *(Corresponding author: Yiwen Zhang.)*

The authors are with the School of Computer Science and Technology, Anhui University, Hefei 230601, China (e-mail: sanglei@ahu.edu.cn; zhanghaiyanahu@gmail.com; wangyuahu@stu.ahu.edu.cn; zhangyiwen@ahu.edu.cn).

Digital Object Identifier 10.1109/TCE.2025.3570293

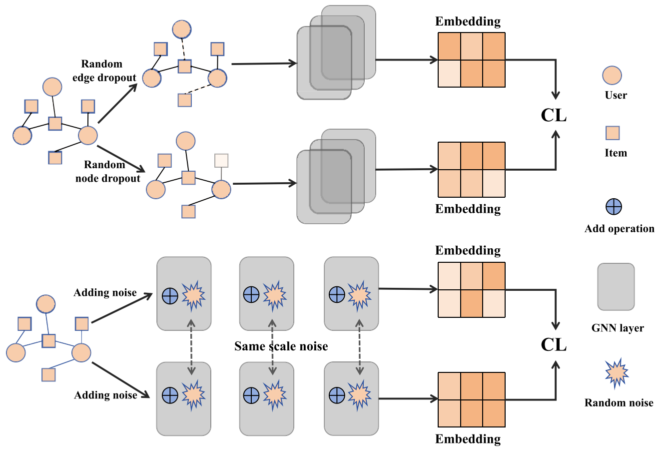


Fig. 1. Graph contrastive learning paradigms with structure and feature data augmentation.

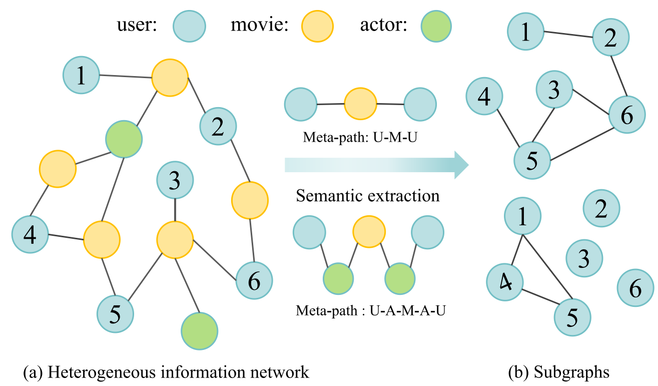


Fig. 2. A toy example of generating subgraphs from a heterogeneous information network based on meta-paths.

principles, contrastive learning optimizes representational coherence across multiple views, thereby extracting more informative supervisory signals. The integration of contrastive learning with Heterogeneous Graph Neural Networks (HGNNs) for unsupervised learning of efficacious user and item embeddings has emerged as a prominent methodological framework. For instance, HeCo, SMIN, and HGCL [26], [27], [28], [29] have leveraged the inherent complex topological structure in HINs, demonstrating the potential of contrastive learning to solve recommendation problems in different scenarios. However, despite these advancements, we believe these techniques are still confronted with the following two key challenges:

CH1: Efficient Generation of High-quality Contrastive Views Although employing contrastive learning has led to better performance in Heterogeneous Graph Neural Network (HGNN)-based recommendation tasks [21], [28], [30]. As shown in Fig. 1, most current contrastive learning models generate different views by node dropping and edge perturbation on the original graph. While such approaches yield relatively effective recommendations, they are not without limitations. The computational complexity associated with node dropping and edge perturbation operations is considerable. Moreover, these techniques potentially compromise valuable graph structural information and introduce extraneous noise, thereby impeding the model's ability to extract salient representational features. Therefore, it is crucial to find a simple yet effective method to construct high-quality contrastive views.

CH2: Effective Differentiation of Node Representations Most current Heterogeneous Graph Neural Network (HGNN)-based contrastive recommenders are subject to the over-smoothing problem, which leads to indistinguishable representations. Common HGNN-based recommendation methods [26], [28], [31], [32] employ smoothing operations on node representations at each layer of the network. As the network depth increases, there is a tendency for node representations to converge, thereby diminishing their discriminative capacity. This issue is further exacerbated by several factors, including suboptimal aggregation of neighborhood information, inadequate graph structural complexity, and non-optimal hyperparameter configurations, all of which can contribute to the over-smoothing problem. The inability to effectively distinguish different representations can hinder the performance of model in capturing the nuanced relationships and differences between users and items, ultimately impacting the quality of the generated recommendations.

In light of the aforementioned limitations, we introduce the SimHGCL (Simple Yet Effective Heterogeneous Graph Contrastive Learning for Recommendation) model. Specifically, we propose a novel approach for extracting user-item submatrices by exploiting the semantic similarity between nodes interconnected via identical meta-paths. As illustrated in Fig. 2, within the Heterogeneous Information Network (HIN), diverse connections such as user-movie and actor-movie exist, representing distinct relationships (e.g., viewing and performing, respectively). Despite the absence of direct user-to-user links, their relationships can be inferred through auxiliary nodes (e.g., movies). The meta-path *user-movie-user*, for instance, can serve as a proxy for user interest similarity. This principle of similarity propagation extends to item-item relationships as well. Leveraging this insight, we generate user/item subgraphs based on meta-paths and subsequently amalgamate the corresponding user/item submatrices into a contrastive matrix. This approach circumvents the necessity for node dropping and edge perturbation operations, thereby addressing the **CH1**. To tackle the **CH2**—the issue of indistinguishable representations stemming from over-smoothing in contrastive recommenders—we introduce a learnable Laplacian matrix exponent. This novel component effectively mitigates the problem of representation homogenization, enhancing the model's capacity to maintain distinct and informative node representations throughout the learning process. Our main contributions are as follows:

- We propose a novel meta-path-based approach for generating contrastive views, simplifying the process and preserving graph integrity by extracting submatrices without structural disruption or noise introduction.
- We introduce a learnable Laplacian matrix exponent in our model, which can effectively solve the problem of

indistinguishable representations caused by over-smoothing in current Graph Neural Network (GNN)-based recommenders.

• We conduct extensive experiments on four public datasets, and the results show that our SimHGCL model outperforms the existing baseline methods in recommendation performance and exhibits a substantial reduction in training time.

II. Related Work

A. GNN-Based Recommendation

Graph Neural Networks (GNNs) have garnered significant attention in recommender systems due to their robust representation learning capabilities, particularly in encoding graph signals through the aggregation and propagation of neighborhood information. This paradigm has led to their widespread adoption across diverse recommendation tasks. Increasingly, research has focused on elucidating GNN interpretability and functional mechanisms. For instance, Defferrard et al. [33] developed efficient localized convolutional filters, He et al. [31] proposed a simplified LightGCN model emphasizing neighbor aggregation, and Wang and Zhang [34] analyzed spectral GNNs, advocating the use of Jacobi bases to enhance representation power. These works, which delve into the underlying structure and operational mechanisms of GNNs, have advanced our understanding of the factors driving GNN performance. Additionally, recent models like DSMN [35], which leverages dynamic routing algorithms and an improved attention mechanism to capture multiple and evolving consumer interests, have significantly enhanced recommender systems in e-commerce environments. Similarly, [36] presents a Web recommender that models consumer behavior using fuzzy logic and periodic patterns for personalized, timely content recommendations. These advancements collectively improve the performance and robustness of recommender systems in complex and dynamic scenarios, enhancing the efficiency and expressiveness of graph-based models and laying a strong foundation for integrating GNNs with diverse models.

B. Contrastive Learning

In recent years, contrastive self-supervised learning has emerged as a prominent paradigm in recommender systems, demonstrating exceptional efficacy across diverse domains. Contrastive learning, a powerful methodological framework, synergistically combines self-supervised signals for data augmentation with the alignment of contrastive representation views. For instance, SGL [21] generates multiple views by perturbing the graph structure, SimGCL [30] enhances contrastive learning by injecting random noise into node representations, and LightGCL [37] simplifies the contrastive learning process through the introduction of singular value decomposition (SVD). These methods aim to improve recommendation performance and computational efficiency by ensuring consistency across different perspectives and integrating collaborative signals. While these methodologies have yielded relatively effective recommendations, the operations they employ—such as node dropout, edge perturbation, and noise addition—present significant challenges. These techniques not only incur high computational complexity but also risk compromising valuable graph structural information and introducing extraneous noise, potentially impeding the model's ability to discern crucial representative information. Consequently, there is an urgent imperative to devise more parsimonious approaches for generating contrastive views that not only sustain but potentially augment performance while simultaneously addressing the aforementioned constraints.

C. Heterogeneous Information Network

Heterogeneous Information Networks (HINs) are characterized by their diverse node and edge typologies, offering a more comprehensive representation of user-item interactions compared to their homogeneous counterparts. Recognizing this potential, researchers have increasingly leveraged HINs to mitigate data sparsity challenges inherent in recommender systems. For example, HGNR [13] enhances representation learning by incorporating auxiliary information, such as user attributes and item categories, while HAN [18] employs attention mechanisms to differentiate the contributions of various node and edge types, addressing challenges like cold start and data sparsity. Although these models have made progress in improving recommendation performance, their ability to capture complex semantic relationships in representations still has room for improvement. To further enhance HIN representation capabilities, recent studies have explored integrating contrastive learning (CL) into the HIN framework. For instance, HGCL [28] models the heterogeneous relational semantics in user-item interactions and uses contrastive learning to facilitate knowledge transfer across views, enabling the model to better capture cross-semantic collaborative signals. While HGCL has demonstrated significant improvements in recommendation accuracy, its complex contrastive view generation process leads to increased computational time, requiring further optimization to improve efficiency.

III. Preliminaries

A. Graph Collaborative Filtering for Recommendation

In the context of Graph Collaborative Filtering (GCF), let us define the sets corresponding to users u and items i as $\mathcal{U}$ and $\mathcal{I}$, respectively. The matrix $R \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$ shows user-item interactions, where $R_{ui} = 1$ indicates that user u interacted with item i, and $R_{ui} = 0$ denotes there was no interaction. Since the user-item interactions R can be modeled as a bipartite graph, the corresponding adjacency matrix A can be formulated as: $A = \begin{bmatrix} 0 & R \\ R^\top & 0 \end{bmatrix}$, where $A \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{I}|)\times(|\mathcal{I}|+|\mathcal{U}|)}$. This setup can be viewed as a link prediction problem between user and item nodes, where the goal is to optimize the randomly initialized user and item embeddings based on the observed user-item interactions.

Let the embedding matrix for users and items at the l-th layer be $E^{(l)} \in \mathbb{R}^{(\mathcal{U}+\mathcal{I})\times T}$, where T represents the embedding dimension. The Graph Collaborative Filtering (GCF) framework employs a multi-layer GCN on the adjacency matrix A to encode high-order collaborative signals into the embeddings.

TABLE I
NOTATIONS AND EXPLANATIONS

Notation	Explanation
$\mathcal{U}, \mathcal{I}$	The sets of users (items).
$R \in \mathbb{R}^{\lvert U \rvert \times \lvert I \rvert}$	The user-item interaction matrix.
A	The adjacency matrix.
$E^{(l)}$	The users/items embedding matrix at layer l.
D	The diagonal degree matrix.
σ	The activation function.
$e_u(e_i)$	The final user (item) embedding.
$\hat{y}$	The likelihood score.
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}, \phi, \varphi)$	The heterogeneous information network.
$\mathcal{A}(\mathcal{R})$	The sets of node (edge) types.
$\phi(\varphi)$	The node (edge) type mapping function.
$G_U^{\rho_k^u}(G_I^{\rho_k^i})$	The user (item) interaction subgraph.
$\rho_k^u(\rho_k^i)$	The user (item) meta-path.
$U_k(I_K)$	The user (item) interaction submatrix.
H_k	The k-th contrast matrix.
γ	The tunable parameter.
$E_{uu}^1, E_{uu}^2(E_{ii}^1, E_{ii}^2)$	Two sets of user (item) embeddings.
$e'(e'')$	Each node in E_{uu}^1 (E_{uu}^2).
τ'	The temperature coefficient.

The embedding generation process via the graph convolutional layer can be expressed as:

$$E^{(l)} = \left(D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right) E^{(l-1)}, \tag{1}$$

where D is the diagonal degree matrix derived from the adjacency matrix A. After obtaining the user and item embeddings from all layers, different GNN models may adopt various methods to produce the final embeddings. For instance, the LightGCN model generates the final embedding by averaging the outputs from all layers. Using the final the user embedding e_u and item embedding e_i, the interaction probability between user u and item i can be calculated as:

$$\hat{y} = \sigma\left(\mathbf{e}_u^\top \mathbf{e}_i\right), \tag{2}$$

where, σ denotes the activation function used. $\hat{y}$ is the likelihood score of an interaction between user u and item i, with higher values indicating a greater probability of interaction.

B. Definitions

Heterogeneous Information Networks (HINs) [26] serve as a sophisticated network structure for depicting and analyzing data with diverse entity types and their multiple relationships. Unlike traditional homogeneous networks, HINs are more structurally intricate, supporting a wide variety of entities and their interconnections, which enables them to convey richer semantic information. The applications of HINs span various domains, including social network analysis, recommender systems, and bioinformatics. For instance, in Fig. 2, the HIN can depict the various relational connections between users, movies, and actors in a movie recommender system.

Here is the definition of a heterogeneous information network (HIN): $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}, \phi, \varphi)$, where $\mathcal{V}$ and $\mathcal{E}$ represent the sets of nodes and edges, respectively. $\mathcal{A}$ and $\mathcal{R}$ are the sets of node types and edge types, and in a HIN, $\mathcal{A} + \mathcal{R} > 2$. The type mappings for nodes and edges are $\mathcal{V} \xrightarrow{\phi} \mathcal{A}$ and $\mathcal{E} \xrightarrow{\varphi} \mathcal{R}$, where ϕ and φ denote the node type mapping function and edge type mapping function, respectively. A meta-path refers to a path in the HIN that connects two nodes through a sequence of edges. As illustrated in Fig. 2, *U-M-U* is a meta-path connecting two user nodes by traversing through a node of type '*M*'. A meta-path can include multiple types of edges and can be defined as $\mathcal{A}_1 \xrightarrow{\mathcal{R}_1} \mathcal{A}_2 \xrightarrow{\mathcal{R}_2} \ldots \xrightarrow{\mathcal{R}_l} \mathcal{A}_{l+1}$. This represents a meta-path connecting $\mathcal{A}_1$ and $\mathcal{A}_{l+1}$

C. Problem Formulation

Data sparsity is a ubiquitous challenge in various recommendation scenarios [1], [15], [29], [38]. By combining GNNs and HINs to explore auxiliary information, the problem of sparse user-item historical interactions can be effectively alleviated [39], [40], [41], [42], [43]. However, the auxiliary information often also suffers from data sparsity issues. To enrich the auxiliary information, numerous works in the field of recommender systems have introduced contrastive learning, constructing multiple enhanced views through operations such as node perturbation, edge dropping, and random walks. This approach effectively alleviates the problem of sparse auxiliary information. Nevertheless, the operations of node perturbation, edge dropping, and random walks to construct contrastive views not only have high complexity, but also disrupt the original graph structure and introduce noise. Moreover, most GNN-based contrastive recommenders are subject to the oversmoothing problem, leading to indistinguishable representations.

Therefore, we propose a simple contrastive view construction method for HINs that does not destroy the original graph structure or introduce noise, mainly based on user/item meta-paths. Formally, given a HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}, \phi, \varphi)$, we first obtain the user interaction subgraph $G_U^{\rho_k^u}$ using the user meta-path (e.g., *U-M-A-M-U*) ρ_k^u, and generate the corresponding user interaction matrix U_k. Similarly, we obtain the item subgraph $G_I^{\rho_k^i}$ and item interaction matrix I_k using the item meta-path ρ_k^i. Users connected by a specific meta-path in the HIN have similar interests to a certain extent. Similarly, items connected by a meta-path may have certain common attributes. Therefore, we can incorporate the submatrices corresponding to the generated user interaction subgraph $G_U^{\rho_k^u}$ and item interaction subgraph $G_I^{\rho_k^i}$ to construct the enhanced contrastive views. Furthermore, we propose a learnable Laplacian matrix exponent, which can effectively solve the problem of indistinguishable representations caused by over-smoothing.

IV. THE PROPOSED MODEL:SIMHGCL

In this section, we introduce the SimHGCL model, a novel contrastive learning framework for heterogeneous information networks (HINs). SimHGCL consists of three key components: Simplified Contrast View Generation, GCN with Adjustable Laplacian Matrix Exponent, and Model Optimization, as illustrated in Fig. 3.

A. Simplified Contrast View Generation

Most current contrastive learning models generate different views by employing node dropout and edge perturbation on

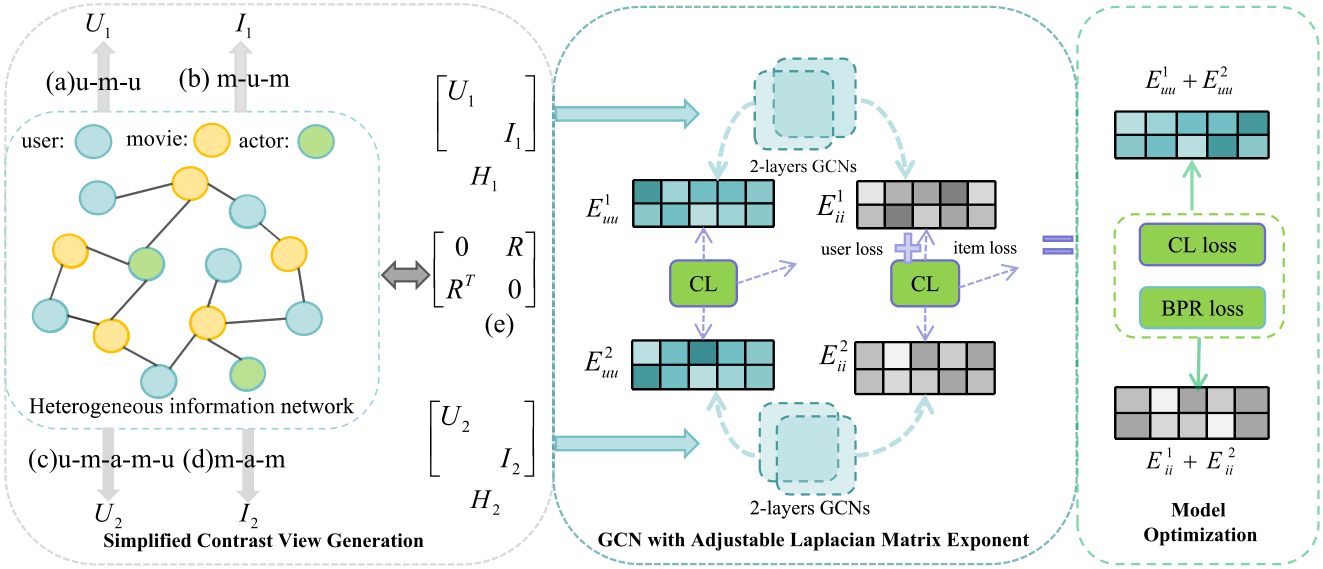


Fig. 3. U_1, I_1, U_2, I_2 correspond to the submatrices generated from the Heterogeneous Information Network (HIN) based on meta-paths (a), (b), (c), (d). (e) corresponds to the user-item interaction matrix of the original HIN. H_1 and H_2 represent the contrastive matrices obtained by incorporating different submatrices. SimHGCL consists of three modules: **Simplified Contrast View Generation** does not require node dropout or edge perturbation operations, while **GCN with Adjustable Laplacian Matrix Exponent** alleviates the indistinguishability issue caused by the over-smoothing problem.

the original graph. These methods not only exhibit high computational complexity but may also disrupt the graph structure, introducing noise. Therefore, in this subsection, we propose a simple yet effective Simplified Contrast View Generation method for constructing contrastive views to address **CH1: Efficient Generation of High-quality Contrastive Views**.

We first present our method for generating contrastive views. As shown in Fig. 2, first, based on different user/item meta-paths ρ_k^u (ρ_k^i), we extract the user/item subgraphs $G_U^{\rho_k^u}$ ($G_I^{\rho_k^i}$) from the original user-item interaction graph [13]. Then, as shown in Fig. 3, we combine the submatrices U_k (I_k) corresponding to the user/item subgraphs $G_U^{\rho_k^u}$ ($G_I^{\rho_k^i}$) to form the contrastive matrix H_k.

$$H_k = \begin{bmatrix} U_k & 0 \\ 0 & I_k \end{bmatrix}, \tag{3}$$

where H_k is the k-th contrast matrix. U_k is the interaction matrix corresponding to the user subgraph $G_U^{\rho_k^u}$ generated by the k-th user-based meta-path ρ_k^u. I_k is the interaction matrix corresponding to the item subgraph $G_I^{\rho_k^i}$ generated by the k-th item-based meta-path ρ_k^i. To improve the recommendation performance through contrastive learning, this paper learns the embeddings of the two H_k and contrasts them, i.e., $k = 2$.

B. GCN With Adjustable Laplacian Matrix Exponent

Most current HGNN-based contrastive recommenders are subject to the oversmoothing problem, which leads to indistinguishable representations. Moreover, HGNN-based recommendation methods employ smoothing operations on node representations at each layer of the network. As the network depth increases, there is a tendency for node representations to converge, thereby diminishing their discriminative capacity. Therefore, in this subsection, we introduce a GCN with Adjustable Laplacian Matrix Exponent setting to tackle **CH2: Effective Differentiation of Node Representations**.

1) Embedding Learning: We use GCNs to learn user-item embeddings from the matrixs. After generating the contrast matrix H_k, we first give the Laplacian matrix of $\mathcal{L}^c$ of H_k:

$$\mathcal{L}^c = D^{\gamma} H_k D^{\gamma}, \tag{4}$$

where, D represents the diagonal matrix. The eigenvalues of the symmetrically normalized Laplacian matrix lie within the interval [0, 2], which better preserves both local and global structural information of the graph. The smallest eigenvalue is always 0, corresponding to a constant eigenvector, facilitating the identification of graph connectivity. By normalizing the degree, the influence of highly connected nodes on clustering and other analyses is mitigated, thereby enhancing the robustness of the results. Unlike conventional approaches, we introduce a tunable parameter γ as the exponent of the diagonal matrix D, instead of using the default $-\frac{1}{2}$. This effectively alleviates the problem of indistinguishable node representations caused by over-smoothing, which we discuss in more detail in the next section.

By using the graph Laplacian matrix $\mathcal{L}^c$, the embedding update function for graph convolution can be derived:

$$E^{(l)} = \sigma\left((\mathcal{L}^{\mathbf{c}} + I)E^{(l-1)}W_o^{(l)} + \mathcal{L}^{\mathbf{c}}E^{(l-1)} \odot E^{(l-1)}W_n^{(l)}\right), \tag{5}$$

where $E^{(l)}$ is the result of concatenating all user and item embeddings after applying (l) graph convolution layers, with $\sigma(\cdot)$ as the sigmoid activation function and I as the identity matrix. The matrices $W_o^{(l)}$ and $W_n^{(l)}$ are trainable weights, where $W_o^{(l)}$ retains each user and item embedding's original data, while $W_n^{(l)}$ gathers information from adjacent nodes.

2) Variable Laplacian Index: In the fields of graph theory and GCNs, the normalized Laplacian matrix is commonly utilized to ensure the stability and effectiveness of feature

Algorithm 1 The SimHGCL Learning Algorithm

Input: User interaction subgraph $G_U^{\rho_k^u}$, item interaction subgraph $G_I^{\rho_k^i}$, maximum training epochs E and parameters required for the training process

Output: Trained node embeddings

1: Initialize all parameters;
2: **for** epoch in $\{1, 2, \ldots, E\}$ **do**
3: Obtain user and item embeddings $E_{uu}^1, E_{uu}^2, E_{ii}^1, E_{ii}^2$ using 2-layer LightGCN via interaction subgraph $G_U^{\rho_k^u}$, $G_I^{\rho_k^i}$, applying Eq. (4);
4: Train E_{uu}^1, E_{uu}^2, E_{ii}^1, E_{ii}^2 by contrastive learning loss according to Eq. (7);
5: Optimize BPR loss $\mathcal{L}_{BPR}$ according to Eq. (13);
6: Joint optimization of SimHGCL via (14);
7: **end for**
8: **return** all parameters and user and item embeddings;

propagation. A widely adopted formulation of the normalized Laplacian matrix is:

$$\hat{L} = \hat{D}^{-\frac{1}{2}}\tilde{A}\hat{D}^{-\frac{1}{2}}, \tag{6}$$

where $\tilde{A}=A+I$, $\hat{D}=D+I$, A and D denote the adjacency matrix and diagonal matrix, respectively, and I refers to the identity matrix. The use of the $-\frac{1}{2}$ exponent in the degree matrix $\hat{D}$ confers three key benefits: (1) balancing node influence, (2) ensuring symmetry, and (3) normalizing feature propagation.

While the fixed degree matrix coefficient of $-\frac{1}{2}$ provides the aforementioned advantages, it can also limit the model's flexibility in controlling the range and influence of feature propagation. Furthermore, different graph structures, due to their topological properties and node distribution characteristics, may benefit from varying degree matrix coefficients. The essence of the oversmoothing problem is that the high power of the normalization matrix causes node features to converge to similar values, dominated by the principal eigenvector. By tuning γ: When $\gamma < \frac{1}{2}$, low-degree nodes are emphasized, preserving local feature differences and mitigating oversmoothing; When $\gamma > \frac{1}{2}$, high-degree nodes dominate, but an appropriate choice of γ balances global and local information. Therefore, in SimHGCL, we propose a variable degree matrix coefficient setting, as expressed in Eq. (4). The variable degree matrix coefficient can bring the following advantages to the model: (1) flexible control over the range and influence of feature propagation by adjusting the exponent value, (2) mitigate the over-smoothing problem by tuning the degree of feature smoothing, and (3) improved adaptability to specific graph structures, leading to enhanced training stability.

3) InfoNCE-Based Contrastive Loss: Contrastive Learning (CL) is a paradigm in unsupervised and self-supervised learning, grounded in the Information Noise-Contrastive Estimation (InfoNCE) principle, also known as InfoMax. Its core principle optimizes feature representations by maximizing Normalized Mutual Information (NMI). The implementation of CL typically involves multiple stages: first, a neural network architecture (often an encoder or convolutional neural network) extracts high-dimensional feature representations from input data. Positive sample pairs are then constructed, typically through different views or augmentations of the same instance, while negative sample pairs are derived from distinct samples or heterogeneous categories within the same dataset.

SimHGCL adopts Random Negative Sampling, where negative samples are uniformly selected from all items, excluding those in the user's positive sample list. This method is computationally efficient but may introduce false negatives, potentially weakening the model's ability to distinguish user-item interactions. Hard negative sampling selects un-interacted items with high similarity to positive samples, enhancing fine-grained preference differentiation, though at a higher computational cost. Distribution-based negative sampling chooses negatives based on item popularity, reducing false negatives and improving mainstream interaction discrimination, but it may overlook long-tail items. Adaptive negative sampling dynamically adjusts negative selection, focusing on challenging samples based on training dynamics, suitable for multi-interest user scenarios, yet complex to implement. We argue that false negatives in sparse datasets may cause the model to erroneously push away potential positive samples. To address this, we propose exploring hybrid sampling strategies (e.g., combining random and hard negative sampling) or regularization techniques (e.g., uniformity loss) in future work to mitigate this issue.

C. Model Optimization

In this subsection, we introduce the optimization method for the proposed SimHGCL model and present the overall loss function along with its time complexity.

1) Loss Function: H_1 and H_2 are processed through a GCN to generate two sets of user embeddings E_{uu}^1 and E_{uu}^2, and two sets of item embeddings E_{ii}^1 and E_{ii}^2. Each node in E_{uu}^1 is represented as e', and each node in E_{uu}^2 is represented as e''. We employ a contrastive learning loss based on InfoNCE [44] to train these two sets of embeddings. The user-based contrastive learning loss, $\mathcal{L}_{\text{user}}$, is defined as follows:

$$\mathcal{L}_{\text{user}} = \sum_{i\in\mathcal{B}} -\log \frac{\exp\left(s\left(\mathbf{e}_i', \mathbf{e}_i''\right)/\tau'\right)}{\sum_{j\in\mathcal{B}} \exp\left(s\left(\mathbf{e}_i', \mathbf{e}_j''\right)/\tau'\right)}, \tag{7}$$

where i, j are users/items in the sampled batch $\mathcal{B}$. $s(\cdot, \cdot)$ denotes the inner product similarity, and τ' stands for the temperature coefficient. Formally, for node $i \in \mathcal{B}$, the gradient of the contrastive learning loss w.r.t. the representation e_i' is given as follows:

$$\frac{\partial \mathcal{L}_{cl}^{\text{user}}(i)}{\partial \mathbf{e}_i'} = \frac{1}{\tau' \|\mathbf{e}_i'\|}\left(\mathbf{c}(i) + \sum_{j\in\mathcal{B}\setminus\{i\}} \mathbf{c}(j)\right), \tag{8}$$

where $\mathcal{L}_{cl}^{\text{user}}(i)$ represents the individual term for node i as defined in Equation (7); $j \in \mathcal{B} \setminus \{i\}$ represents another node serving as the negative view for node i. The terms $c(i)$ and

$c(j)$ denote the contributions of the positive node i and the negative node j, respectively, to the gradient of $\mathbf{e}'_i$:

$$c(i) = \left(s''_i - \left(s'^T_i s''_i\right)s'_i\right)^T (P_{ii} - 1), \tag{9}$$

$$c(j) = \left(s''_j - \left(s'^T_i s''_j\right)s'_i\right)^T P_{ij}, \tag{10}$$

where $P_{ij} = \frac{\exp(s'^T_i s''_j/\tau')}{\sum_{j\in\mathcal{U}} \exp(s'^T_i s''_j/\tau')}$, $s'_i = \frac{\mathbf{e}'_i}{||\mathbf{e}'_i||}$ and $s''_i = \frac{\mathbf{e}''_i}{||\mathbf{e}''_i||}$ are the normalized representations of node i in different views; similar notation applies to node j. Next, we focus on the contribution of the negative node j (cf. Equation (10)), the L_2 norm of which is proportional to the following term:

$$||c(j)||_2 \propto \sqrt{1 - \left(s'^T_i s''_j\right)^2 \exp\left(s'^T_i s''_j/\tau'\right)}. \tag{11}$$

Similarly, the item-based contrastive loss, $\mathcal{L}_{\text{item}}$, can be computed. The total contrastive loss is:

$$\mathcal{L}_{\text{cl}} = \mathcal{L}_{\text{user}} + \mathcal{L}_{\text{item}}. \tag{12}$$

The proposed model employs the pairwise loss function of Bayesian Personalized Ranking (BPR) [45] to optimize the recommendation task. The BPR loss function incorporates both the items that each user u_i has interacted with v_{p_i} (positive items) and the items that the user has not interacted with v_{n_i} (negative items). Specifically, the model seeks to maximize the prediction score as follows:

$$\mathcal{L}_{bpr} = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\sigma\left(\sum_{i=1}^{d} u_i \cdot v_{p_i} - \sum_{i=1}^{d} u_i \cdot v_{n_i}\right)\right), \tag{13}$$

where $\log(\cdot)$ represents the logarithm function and σ denotes the sigmoid function.

The total loss function combines the BPR loss, contrastive learning loss, and a regularization loss, as follows:

$$\mathcal{L} = \mathcal{L}_{bpr} + \lambda_{\text{cl}}\mathcal{L}_{cl} + \lambda_{\text{reg}}\mathcal{L}_{reg}, \tag{14}$$

the hyperparameter λ_{reg} determines the weight of the regularization term, while the value of λ_{cl} defines the significance of the contrastive learning term in the overall model.

2) Model Complexity: We conducted a comprehensive analysis of the time complexity of the proposed SimHGCL model. As a representative example, we considered the batch processing time complexity commonly employed in contrastive learning approaches. The time complexity involved in normalizing the original adjacency matrix is $\mathcal{O}(2|E|)$, where E indicates the edge count in the graph. For graph convolution, the computational cost is $\mathcal{O}(4|E|Ld)$, where L indicates the number of layers and d signifyies the embedding size in GNNs. The Bayesian Personalized Ranking (BPR) loss function, used for recommendation, has a computational complexity of $\mathcal{O}(2Bd)$, where B stands for the batch size. Additionally, the contrastive loss incurs a complexity of $\mathcal{O}(Bd + BMd)$, where M refers to the number of nodes within the batch. Combining these individual components, the overall time complexity of the SimHGCL model is $\mathcal{O}(2|E| + 4|E|Ld + 3Bd + BMd)$. Table III compares the computational complexities of SimHGCL, LightGCN, and SGL. SimHGCL exhibits a significantly reduced time complexity relative to SGL, which also adopts a classic contrastive learning approach.

TABLE II
EXPERIMENT DATA STATISTICS

Dataset (Sparsity)	#User	#Item	#Relations	Meta-paths
Yelp (99.91%)	16239	14284	5	UU, UBU $BCIB, BCAB$
Amazon (98.85%)	6170	2753	5	$UIU, UIBIU$ IBI, ICI
Doubanmovie (99.37%)	13367	12677	6	UU, UGU MUM, MDM
Movielens (95.81%)	6040	3952	5	UMU, UOU MUM, MGM

TABLE III
TIME COMPLEXITY COMPARISON

Component	LightGCN	SGL-ED	NCL	SimHGCL
Adj	$\mathcal{O}(2\|E\|)$	$\mathcal{O}(2\|E\| + 4\rho\|E\|)$	$\mathcal{O}(2\|E\|)$	$\mathcal{O}(2\|E\|)$
GC	$\mathcal{O}(2\|E\|Ld)$	$\mathcal{O}((2+4\rho)\|E\|Ld)$	$\mathcal{O}(2\|E\|Ld)$	$\mathcal{O}(4\|E\|Ld)$
BPR Loss	$\mathcal{O}(2Bd)$	$\mathcal{O}(2Bd)$	$\mathcal{O}(2Bd)$	$\mathcal{O}(2Bd)$
CL Loss	-	$\mathcal{O}(Bd + BMd)$	$\mathcal{O}(Bd + BMd)$	$\mathcal{O}(Bd + BMd)$

V. EXPERIMENT

We carry out thorough experiments on four public recommendation datasets in this section to confirm the efficacy of SimHGCL and to respond to the following inquiries:

RQ1: How does SimHGCL compare to the cutting-edge contrastive learning methods based on HGNNs?

RQ2: How does our model perform against data sparsity, popularity bias?

RQ3: Can SimHGCL effectively alleviate the problem of high time complexity in the contrastive view generation process?

RQ4: Does each component design of SimHGCL make a positive contribution to the recommendation results?

RQ5: What impact do various parameter configurations have on its performance?

A. Experimental Setup

In this subsection, we provide an overview of the datasets and evaluation metrics utilized in the experiments, as well as the baseline models and parameter settings.

1) Datasets and Metrics: The methods were evaluated using four real-world datasets: Yelp [46] for business, Movielens-1M[1] and Douban[2] for movies, and Amazon[3] for products, aiming to predict user interactions with different entities such as businesses, movies, and products. Table II displays the statistical summary for the four datasets. Additionally, we employed two well-known metrics, Recall@K and NDCG@K, to evaluate all methods, and used the P-value [6], [47], [48] to assess the statistical significance of the results. It serves as a statistical measure for significance testing because it quantifies the extent of inconsistency between the observed results and the null hypothesis, helping to determine the statistical significance of the results.

[1]https://grouplens.org/datasets/movielens/
[2]https://m.douban.com/
[3]http://jmcauley.ucsd.edu/data/amazon/

2) Baselines: SimHGCL's performance was compared against that of the following eight models:

- *HERec* [49] employs a meta-path guided random walk strategy to form node sequences, facilitating effective embedding of nodes in HINs.
- *HAN* [18], as a GAT-based architecture, integrates node-level and semantic-level attention mechanisms to assess the significance of nodes and their neighbors through meta-paths, optimizing semantic information across different meta-paths.
- *LightGCN* [31] retains only the core neighborhood aggregation component from GCNs, aiming to enhance recommendation performance by a simplified model architecture.
- *HGT* [12] introduces the Heterogeneous Graph Transformer architecture, leveraging node-type and edge-type dependent attention mechanisms and the HGSampling algorithm to achieve efficient modeling and representation learning for Web-scale heterogeneous graphs.
- *HeCo* [26], as an unsupervised CL method, designs network patterns and meta-path views as two views, and effectively learns node embeddings through a cross-view contrastive mechanism.
- *SMIN* [27], as an unsupervised CL method, utilizes a self-supervised metagraph learning approach to effectively integrate information from social networks, and captures the complex heterogeneous relationships through a meta-path guided HGNN module.
- *NCL* [20] introduces the concepts of structural and semantic neighbors to enrich the contrastive learning process, leveraging high-order paths in the graph structure and similarities in semantic space to augment representation learning.
- *HGCL* [28], as an unsupervised CL method, enhances the modeling capability of interactions between users and items by integrating semantic information from heterogeneous relationships and leveraging contrastive learning techniques.
- *RecDCL* [50] proposes a novel dual-contrastive learning recommendation framework to eliminate redundant solutions in the user-item representation learning process.

3) Parameter Settings: The experiments were conducted based on the PyTorch implementation. In both the SimHGCL model and each baseline model, the embedding dimension was fixed at 64 to prevent overfitting and excessive computational cost. For the SimHGCL model, the parameters were set as follows. Optimal values were determined in a standard setting: the learning rate ranged from 0.001 to 0.005, the contrastive loss scaling factor was between 0.01 and 0.05, the L2 regularization term was set to 0.0001 across all four datasets, the temperature parameter varied between 0.1 and 0.5, and the exponent adjustment for the diagonal matrix of the Laplacian ranged from $[-0.9, -0.8, \ldots, -0.2, -0.1]$. To balance efficiency and resource usage: the test batch size was set to 300 for all datasets, a two-layer GCN was used across all datasets, and the batch size was set to 4096 for each dataset.

B. Main Experimental Analysis (RQ1)

1) Overall Experimental Performance Comparison: SimHGCL consistently outperforms the eight baseline models across all datasets and metrics. This can be attributed to:

- **Simplified contrastive view generation:** Unlike traditional contrastive learning methods that use computationally expensive techniques (e.g., node dropout, edge perturbation), SimHGCL generates two matrices from the user-item interaction graph using meta-paths. This reduces complexity and preserves the graph structure, avoiding noise from random augmentations, and enhances the model's ability to learn meaningful representations.
- **Learnable Laplacian matrix exponent:** Over-smoothing in GNNs, especially in deeper models, is mitigated by SimHGCL's learnable Laplacian matrix exponent, allowing better control of feature propagation. This helps maintain node embedding uniqueness and improves recommendation quality.
- **Regularization effect of contrastive learning:** Contrastive learning acts as a self-supervised regularization technique that helps the model learn generalizable features, leading to more robust representations and stable performance across metrics.

2) Performance Variations Across Different Datasets: Across the four datasets, SimHGCL shows greater improvements over baseline models on the Douban Movie and Yelp datasets, while the improvements are smaller on the Amazon and Movielens datasets. We believe this may be due to the higher sparsity of the Douban Movie and Yelp datasets, which have a larger number of nodes. SimHGCL's ability to extract user/item relationships based on meta-paths and its adaptive Laplacian matrix may better handle sparse graph data, helping to avoid the over-smoothing problem, thus leading to better performance on these datasets. For denser datasets like Amazon and Movielens, SimHGCL's advantage may be less pronounced, as traditional models (e.g., LightGCN) already perform well on dense graphs.

3) Performance on Different Evaluation Metrics: Performance on Different Evaluation Metrics: The main experimental results show that SimHGCL's improvement on Recall@10 and Recall@20 is greater than its improvement on NDCG@10 and NDCG@20. We believe this phenomenon may be due to the different focuses of Recall and NDCG. Recall primarily measures the model's ability to successfully identify items of interest to the user in the recommendation list, focusing on whether relevant items appear in the results, regardless of their specific positions in the list. In contrast, NDCG not only considers whether relevant items are recommended but also takes into account their positions in the list, with higher scores for items ranked closer to the top. Therefore, NDCG is a more fine-grained metric that reflects the quality of the ranking. Since SimHGCL's contrastive learning objective is mainly focused on distinguishing between positive and negative samples, it is more conducive to improving Recall. Furthermore, the P-value for all datasets is less than 0.001, indicating that our model achieves significant improvements over all the best baselines.

TABLE IV

OVERALL PERFORMANCE COMPARISON BOLD TEXT REPRESENTS THE BEST RESULTS, AND UNDERLINED TEXT DENOTES THE SECOND-BEST. THE 'IMPROVE' VALUE DEMONSTRATES THE RELATIVE GAIN OF SIMHGCL OVER THE SECOND-BEST RESULTS. OUR SIMHGCL SURPASSES THE CURRENT BENCHMARKS

Dataset	Metric	HERec (2018)	HAN (2019)	LightGCN (2020)	HGT (2020)	HeCo (2021)	SMIN (2021)	NCL (2022)	HGCL (2023)	RecDCL (2024)	SimHGCL (Ours)	p-value
Yelp	R@10	0.0459	0.0419	0.0591	0.4070	0.0526	0.0537	0.0613	<u>0.0631</u>	0.0473	**0.0684**	$2.9e^{-7}$
	R@20	0.0771	0.0737	0.0893	0.0784	0.0819	0.0862	0.0933	<u>0.0959</u>	0.0807	**0.1062**	$2.3e^{-6}$
	N@10	0.0383	0.0362	0.0468	0.0352	0.0407	0.0416	<u>0.0518</u>	0.0507	0.0359	**0.0544**	$6.7e^{-5}$
	N@20	0.0473	0.0443	0.0546	0.0468	0.0501	0.0514	<u>0.0609</u>	0.0602	0.0457	**0.0644**	$5.8e^{-5}$
	P@10	0.0148	0.0142	0.0243	0.0156	0.0172	0.0207	<u>0.0245</u>	0.0186	0.0232	**0.0259**	$0.9e^{-4}$
	P@20	0.0123	0.0119	<u>0.0199</u>	0.0128	0.0142	0.0167	0.0196	0.0148	0.0186	**0.0207**	$2.9e^{-4}$
Douban movie	R@10	0.1110	0.1023	0.1141	0.1135	0.1108	0.1119	0.1156	<u>0.1181</u>	0.1139	**0.1326**	$9.0e^{-7}$
	R@20	0.1761	0.1687	0.1779	0.1696	0.1739	0.1738	0.1803	<u>0.1855</u>	0.1778	**0.2050**	$1.9e^{-8}$
	N@10	0.1757	0.1721	0.2011	0.1644	0.1971	0.1842	<u>0.2055</u>	0.2017	0.1387	**0.2201**	$1.1e^{-6}$
	N@20	0.1833	0.1778	0.2003	0.1904	0.1963	0.1856	<u>0.2067</u>	0.2042	0.1429	**0.2210**	$3.9e^{-6}$
	P@10	0.1367	0.1292	<u>0.1579</u>	0.1341	0.1255	0.1325	0.1557	0.1429	0.1338	**0.1681**	$1.4e^{-4}$
	P@20	0.1134	0.1048	<u>0.1374</u>	0.1093	0.0994	0.1083	0.1285	0.1163	0.1065	**0.1396**	$0.6e^{-3}$
Movielens	R@10	0.1441	0.1327	0.1584	0.1469	0.1514	0.1518	0.1486	<u>0.1634</u>	0.1539	**0.1755**	$8.6e^{-7}$
	R@20	0.2300	0.2133	0.2513	0.2271	0.2372	0.2383	0.2338	0.2528	<u>0.2559</u>	**0.2724**	$2.6e^{-7}$
	N@10	0.3607	0.3562	0.3912	0.3457	0.3821	0.3831	0.3789	<u>0.4073</u>	0.3536	**0.4145**	$1.7e^{-5}$
	N@20	0.3504	0.3386	0.3773	0.3584	0.3665	0.3676	0.3623	<u>0.3853</u>	0.3594	**0.4007**	$1.1e^{-6}$
	P@10	0.3356	0.3302	<u>0.3452</u>	0.2941	0.2980	0.2452	0.3322	0.2698	0.3108	**0.3639**	$2.2e^{-3}$
	P@20	0.2693	0.2687	<u>0.2887</u>	0.2421	0.2459	0.2044	0.2760	0.2085	0.2527	**0.3015**	$1.3e^{-3}$
Amazon	R@10	0.0917	0.0874	0.1034	0.0892	0.0993	0.1035	<u>0.1164</u>	0.1091	0.0902	**0.1197**	$6.0e^{-4}$
	R@20	0.1415	0.1391	0.1589	0.1337	0.1517	0.1572	<u>0.1716</u>	0.1682	0.1387	**0.1779**	$4.6e^{-5}$
	N@10	0.0861	0.0826	0.0981	0.0819	0.0941	0.0973	<u>0.1101</u>	0.1071	0.0737	**0.1139**	$7.2e^{-5}$
	N@20	0.1024	0.0992	0.1147	0.0981	0.1104	0.1141	<u>0.1272</u>	0.1252	0.0956	**0.1321**	$4.0e^{-6}$
	P@10	0.0362	0.0358	0.0608	0.0342	0.0262	0.0277	<u>0.0655</u>	0.0359	0.0406	**0.0666**	$0.7e^{-3}$
	P@20	0.0297	0.0284	0.0474	0.0290	0.0215	0.0231	<u>0.0508</u>	0.0278	0.0319	**0.0514**	$1.1e^{-3}$

4) Comparison With Different Types of Baseline Models: Reasons of SimHGCL outperforms different baseline models:

- **Meta-path-based Models (HERec):** SimHGCL outperforms HERec, an earlier meta-path-based model, by generating contrastive views based on meta-paths, avoiding noise from traditional contrastive learning methods. The learnable Laplacian matrix exponent further mitigates over-smoothing, and SimHGCL achieves better results with lower computational complexity.
- **Graph Neural Network-based Models (LightGCN):** SimHGCL surpasses LightGCN, which struggles with over-smoothing and heterogeneous graphs. SimHGCL's learnable Laplacian matrix and meta-path-based views overcome these challenges, leading to better overall performance.
- **Contrastive Learning-based Models (HeCo, RecDCL):** SimHGCL outperforms HeCo and RecDCL, which rely on random augmentations that introduce noise and increase complexity. SimHGCL's meta-path-based views avoid such issues, improving efficiency and performance.
- **Recent Strong Baselines (NCL, HGCL):** SimHGCL outperforms NCL and HGCL, which still face challenges in view generation and over-smoothing. SimHGCL's meta-path-based views and learnable Laplacian matrix solve these problems while maintaining computational efficiency.

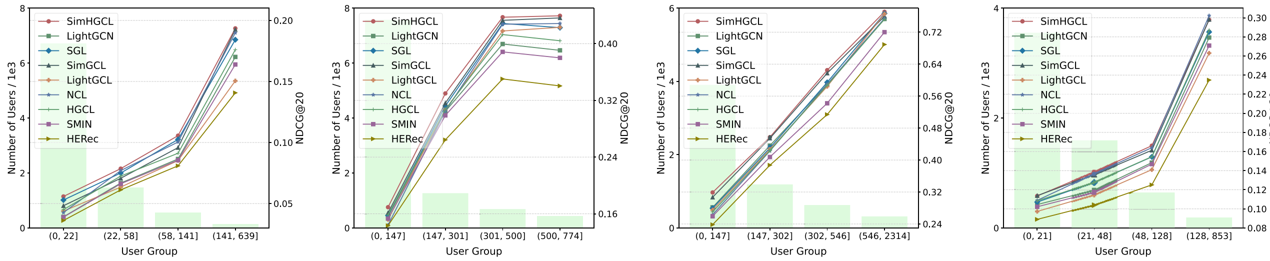
C. Comparison Experiments

1) Performance Comparison w.r.t. Data Sparsity, Popularity Bias (RQ2): Recommender systems commonly face the difficulty of data sparsity, and the ability to effectively alleviate this issue is an important criterion to gauge the efficacy of a model. To determine SimHGCL's effectiveness in managing data sparsity, we tested its performance on various datasets, including Amazon, Yelp, Doubanmovie, and Movielens. We compared SimHGCL against several traditional GNN models and contrastive learning models, such as LightGCN, SGL, and HGCL, on the NDCG@20 metric. The Recall@20 results exhibited similar trends.

To better separate users with different levels of interaction, we classified each dataset into four groups based on interaction frequency. We followed the settings in [16], keeping the total count of interactions in each group approximately equal to ensure that users in each group have sufficient information for training. Using the Amazon dataset as a case in point, the four groups were 0-21, 22-48, 49-128, and 129-853. The number of users in the 0-21 group was as high as 3553, while it in the 129-853 group was only 192. This division ensured that each group's total interactions is around 45,000.

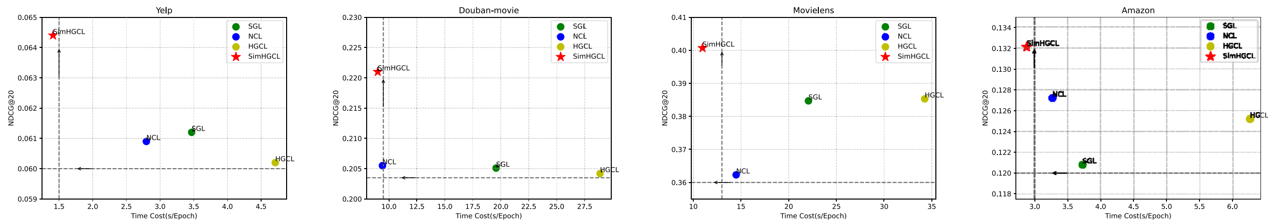
As depicted in Fig. 4, SimHGCL performs as well as or better than the best baseline across all four datasets and their groups. This suggests that SimHGCL can efficiently address the issue of data sparsity. By employing the variable Laplacian index setup we proposed, SimHGCL is able to overcome the challenge of indistinguishable representations due to over-smoothing and perform well in distinguishing representations in sparse environments.

Moreover, the recommender system domain suffers from well-known challenge of popularity bias, where for users



(a) Yelp (b) Doubanmovie (c) Movielens (d) Amazon

Fig. 4. Different sparsity levels are compared in terms of performance. In the background, the bar graph shows the number of users in each group (left y-axis), whereas the line graph represents the highest performance of each method relative to NDCG@20 (right y-axis). The horizontal axis denotes the interaction intervals for each user group.



(a) Yelp (b) Doubanmovie (c) Movielens (d) Amazon

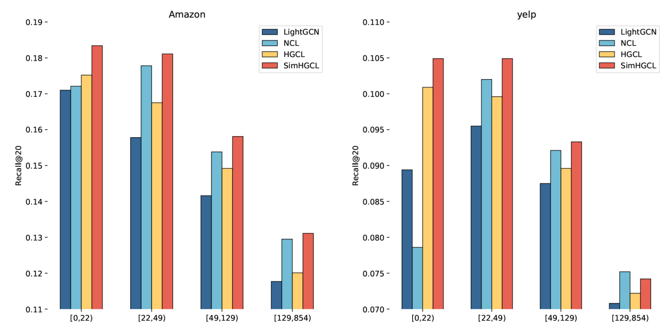
Fig. 5. Comprehensively compare the recommendation performance (y-axis, the higher the better) and training speed (x-axis, the lower the better) of SimHGCL with other benchmark methods.

with fewer interactions, the system is unable to effectively learn their unique interest preferences. As a result, users tend to receive more recommendations for popular items, while less popular items receive fewer recommendations. Fig. 6 exhibits the Recall@20 performance of SimHGCL and three baseline models across multiple sparsity groups. Even in the groups with fewer user interactions, SimHGCL achieves good recommendation results, indicating that our SimHGCL model has a certain degree of resistance to popularity bias.

2) Performance Comparison w.r.t. Complexity (RQ3): As depicted in Fig. 5, we performed an extensive evaluation of the recommendation effectiveness and execution time of SimHGCL compared to three baseline contrastive learning models. The results demonstrate the SimHCGL model surpasses the three baseline models with its quicker runtime and improved NDCG@20 results, with consistent trends observed in the Recall@20 metric. This superior performance can be attributed to the fact that SimHGCL only needs to extract two contrastive matrices based on meta-paths in the HIN, without the need for operations such as edge dropping, node perturbation, and adding noise like other contrastive learning models. Consequently, SimHGCL's complexity is greatly diminished. Moreover, the effectiveness of the simplified contrastive generation approach and the efficiency of the variable Laplacian matrix exponent setting utilized in SimHGCL enable the model to maintain optimal performance.

D. In-Depth Analysis of SimHGCL

In this subsection, we first conduct ablation experiments on the various components of SimHGCL to investigate the contribution of each component to overall performance.



(a) Amazon (b) Yelp

Fig. 6. SimHGCL's ability to alleviate popularity bias in comparison to baselines. The x-axis denotes the interaction intervals for each user group.

Next, we visualize the embeddings of SimHGCL and compare the embedding distributions to explore the sources of performance. Finally, we experiment with three key parameters of SimHGCL to determine the optimal configurations.

1) Model Component Analysis w.r.t. Ablation (RQ4): We conducted a comprehensive ablation study to examine the effectiveness of each key module within the SimHGCL framework. Specifically, we systematically stripped away critical components from the SimHGCL model to assess their individual contributions.

W/o CL: This variant disables the Contrastive Learning (CL) feature by nullifying the CL loss, thereby ignoring the contrastive information between samples during the training.

W/o U: This variant discards the user submatrix U, retaining solely the item submatrix I for contrastive learning.

W/o I: This variant eliminates the item submatrix I, keeping only the user submatrix U for contrastive learning.

TABLE V
ABLATION OF ESSENTIAL PARTS IN SIMHGCL

Data	Yelp		Doubanmovie		Movielens		Amazon	
Metric	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20
W/o CL	0.0965	0.0594	0.1857	0.2100	0.2554	0.3881	0.1737	0.1269
W/o U I	0.0993	0.0625	0.2014	0.2191	0.2716	0.3992	0.1716	0.1262
W/o U	0.1012	0.0635	0.1987	0.2182	0.2723	0.4007	0.1729	0.1274
W/o I	0.1045	0.0638	0.2058	0.2219	0.2714	0.4004	0.1731	0.1275
Exponent of -1/2	0.0922	0.0575	0.1942	0.2106	0.2724	0.4007	0.1719	0.1263
SimHGCL	0.1062	0.0644	0.2050	0.2210	0.2724	0.4007	0.1779	0.1321

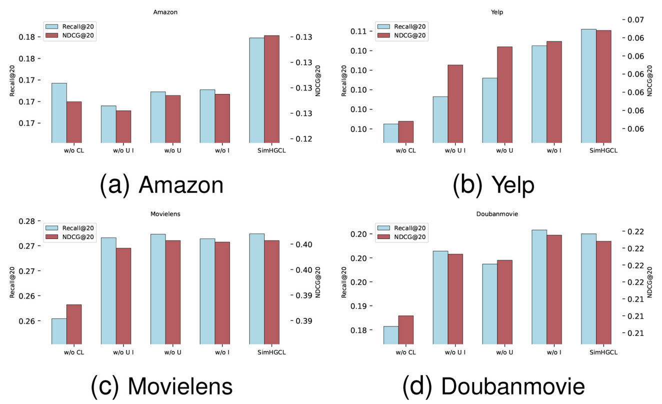


Fig. 7. The performance of ablation experiments conducted on four datasets for Recall@20 and NDCG@20.

W/o U I: This variant maintains solely the initial user-item interaction matrix for contrastive learning.

Table V and Fig. 7 provide a summary of the experimental results for Recall@20 and NDCG@20 regarding these four variants and four datasets. Analyzing the ablation performance, we observe that in the Amazon dataset, all variants with removed modules perform poorly, indicating that the removal of any module affects SimHGCL's efficiency. Similarly, the results of the ablation experiment on the Yelp dataset are analogous, with all variants exhibiting subpar performance when compared to the full SimHGCL model.

Furthermore, the experimental results from the ablation tests reveal that the W/o I variant achieves much higher performance compared to the W/o U variant in the Yelp dataset. This suggests that incorporating the user submatrix U has a greater impact than incorporating the item submatrix I in the Yelp dataset. The data statistics show that the user-to-item ratio in the Yelp dataset is close to 1:1, much lower than the 2:1 ratio in the Amazon dataset. Therefore, we can infer that incorporating the user matrix U has a greater impact than incorporating the item matrix I in the Yelp dataset due to the relatively sufficient item data.

In the Movielens dataset, the full SimHGCL model continues to deliver the best performance. However, the performance gap between the variants is relatively small, and the improvement from retaining the user submatrix U and the item submatrix I is also modest, which we attribute to the relatively low sparsity (95.81%) of the Movielens dataset. In contrast, in the Doubanmovie dataset, the variant that retains only the user submatrix U and removes the item submatrix I exhibits the best performance. We conduct further analysis experiments on this observation for the Doubanmovie dataset in the following sections.

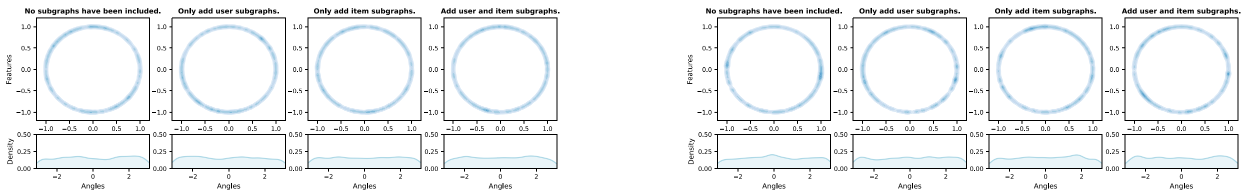
2) Embedding Visualization (RQ4): Based on the experimental settings of SimGCL, we have visualized the embeddings of SimHGCL. As shown in Fig. 8, the embeddings are generated on the Amazon and Doubanmovie datasets under four settings: no subgraph, only user subgraph, only item subgraph, and both subgraphs (In the feature distribution, areas with darker tones have a higher number of points). The visualization analysis demonstrates that the incorporation of subgraph structures can substantially modify the distribution of node embeddings, in contrast to the setting where no subgraphs are utilized.

In the Amazon dataset, the four settings exhibit some clustering of points, but the overall distribution remains relatively uniform, indicating that the model retains the inherent features of the nodes while maintaining strong generalization capability. Notably, the setting that utilizes both the user and item subgraphs performs the best on the Amazon dataset, as it achieves this desirable balance through a more tailored embedding distribution.

On the Doubanmovie dataset, the setting with only the item subgraph used exhibits the desired characteristic of having some clustering points while maintaining a relatively uniform overall distribution, which is consistent with the superior performance of this setting in the ablation study.

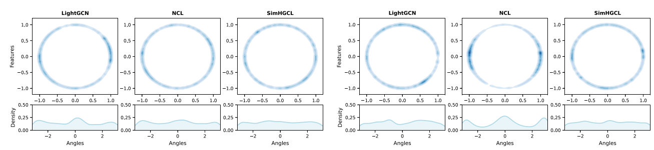
Furthermore, in Fig. 9, we visualize the user embedding distributions of SimHGCL, LightGCN, and NCL on the Yelp and Doubanmovie datasets. Compared to the overly clustered embeddings of LightGCN and NCL, the embeddings of our SimHGCL model are more uniformly distributed, indicating stronger generalization capability. Importantly, SimHGCL retains the clustering points, which can effectively differentiate different nodes, without excessively pursuing uniformity. Additionally, we selected six users from the Amazon dataset and visualized the embedding distributions of these users and the items they interacted with, as shown in Fig. 10. The results demonstrate that, compared to LightGCN, NCL, and HGCL, the SimHGCL model better distinguishes between different user embeddings, while the distance between user embeddings and their corresponding item embeddings is also closer.

3) Case Study (RQ4): We explored the impact of incorporating a user/item submatrix into the user-item historical interaction matrix on the generation of recommendation lists using the MovieLens-1M dataset. Taking User1459 as an example, the comparison results are shown in Fig. 11. As illustrated, User1459's interaction history (with ratings ≥ 4) shows a preference for movies in the Comedy, Musical, and Drama



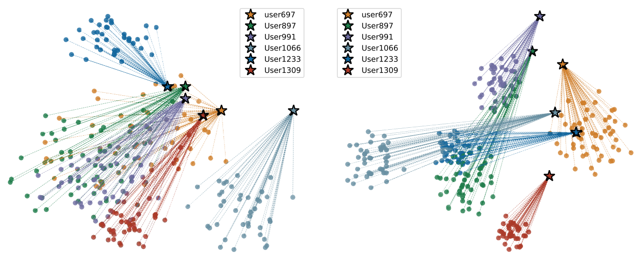
(a) Layout of user embeddings from Amazon dataset. (b) Layout of user embeddings from Doubanmovie dataset.

Fig. 8. Visualize the impact of user and item subgraphs on the distribution of user embeddings.

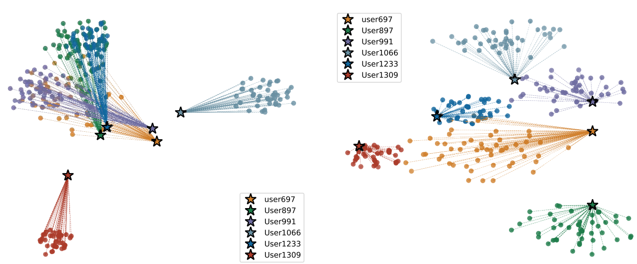


(a) Yelp (b) Doubanmovie

Fig. 9. Comparison of user embedding distributions between SimHGCL and the baseline.



(a) LightGCN (b) NCL



(c) HGCL (d) SimHGCL

Fig. 10. Embedding visualization of users (stars) and interacted items (circles) based on different encoding methods.

genres. Consequently, in Fig. 11(b), the recommendation list generated using only the user-item interaction matrix included movies such as *Pushing Tin* (Comedy), *Yellow Submarine* (Musical), and *Music of the Heart* (Drama). After comparing this with the actual user interactions in Fig. 11(d), we found that these three movies indeed match the user's genuine preferences. However, due to the sparse interactions with movies in the "Sci-Fi | Thriller" genre, the recommendation list in Fig. 11(b) missed the movie *Predator* (Sci-Fi | Thriller), which the user also liked.

In contrast, the recommendation list generated in Fig. 11(c), which incorporates the user/item submatrix, included all the aforementioned movies and effectively utilized item similarity to recommend *Predator* (Sci-Fi | Thriller), even in the face of sparse data for this genre. Additionally, the list in Fig. 11(c)

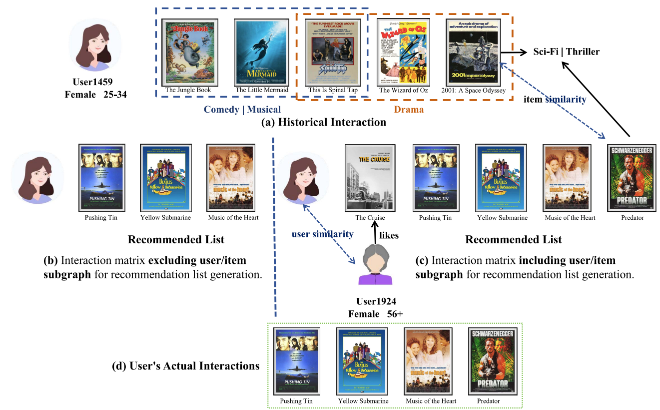


Fig. 11. Comparison of recommendation lists before and after incorporating the user/item submatrix.

also suggested *The Cruise* (Documentary), a movie favored by User1924, who has similar preferences to User1459. This demonstrates that the recommendation approach incorporating the user/item submatrix focuses more on the correlations between user preferences, rather than being limited to the user's own interaction history, thereby helping to uncover potential interests.

4) Parameter Analysis (RQ5): We conducted a comprehensive parameter analysis to investigate the impact of three key hyperparameters on the model's performance: embedding dimension, GCN layer count, and learning rate, and conducted parameter analysis experiments on four datasets. For the embedding dimension, we set it to [16, 32, 64, 128, 256]. From Fig. 12, the Recall@20 and NDCG@20 metrics generally increased with larger embedding dimensions, except for Amazon, which reaches the maximum value at 128 dimensions for the NDCG@20 metric. This phenomenon arises because a larger embedding dimension enables the model to more effectively capture and represent the complex relationships within the data. However, as the embedding dimension increases, the computational complexity and memory usage often also increase. Moreover, an excessively large embedding dimension may lead to overfitting and the introduction of noise. To reduce the complexity of the experiment and avoid overfitting, as well as to allow for comparison with conventional studies, we configured the embedding dimension to 64.

Regarding the GCN layer count, we compared 1 to 4 layers. From Fig. 12, the results indicate that except for the Amazon dataset, which achieved the maximum value in both metrics with a 1-layer GCN, the other three datasets reached their optimal performance with a 2-layer GCN. This suggests that

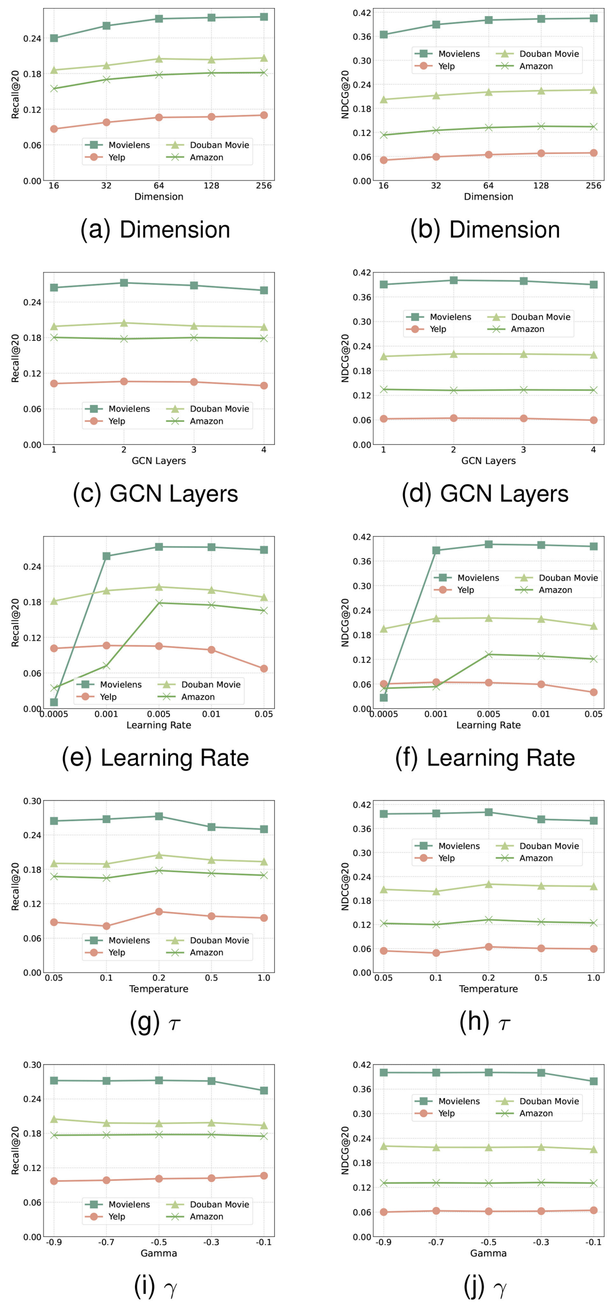


Fig. 12. The Recall@20 and NDCG@20 score w.r.t. Dimension, GCN Layers, Learning Rate, Temperature Coefficient τ and Laplacian Coefficient γ.

a sufficient number of GCN layers is necessary to effectively capture the representational information, while a large number of layers can increase the model's complexity. Therefore, we set the GCN layer count to 2.

For the learning rate, we set it to [0.0005, 0.001, 0.005, 0.01, 0.05]. From Fig. 12, the Yelp dataset achieved its optimal performance at a learning rate of 0.001, while the remaining three datasets reached their peak values at a learning rate of 0.005. A learning rate that is too large may skip the optimal solution or even fail to converge, while a learning rate that is too small may result in a slow convergence speed or even a local optimum. Therefore, we set the learning rate for the Yelp dataset to 0.001 and the learning rate for the remaining datasets to 0.005.

Regarding the temperature coefficient, we conducted experiments with values of [0.05, 0.1, 0.2, 0.5, 1]. As shown in Figure 12, all four datasets achieved their optimal performance at a temperature coefficient of 0.2. A large temperature coefficient may make the model overly sensitive to noise, thereby affecting its generalization ability, while a small temperature coefficient may weaken the contrastive learning signal, resulting in poor learning outcomes. Therefore, we set the temperature coefficient to 0.2 for all experiments.

Finally, regarding the Laplacian index, we conducted experiments with values of $[-0.9, -0.7, -0.5, -0.3, -0.1]$. As shown in Figure 12, each of the four datasets has an optimal value for the index, but there is no clear linear relationship between the index value and performance. In response to this, we provide a detailed theoretical analysis and explanation in Section V-B. (GCN with Adjustable Laplacian Matrix Exponent).

VI. Conclusion

In this study, we propose a simple and effective heterogeneous graph contrastive learning recommendation model, called SimHGCL. Our approach generates two matrices based on meta-paths from the user-item interaction graph, without relying on common techniques in contrastive recommender systems such as node dropout or edge perturbation. Moreover, to address the issue of indistinguishable representations caused by over-smoothing in contrastive recommenders, we introduce a learnable Laplacian matrix exponent, which effectively mitigates this problem. To validate the effectiveness of the SimHGCL model, we conducted extensive experiments on four real-world datasets. The results demonstrate that our method significantly outperforms state-of-the-art baselines in terms of normalized discounted cumulative gain (NDCG) and recall metrics. Notably, the SimHGCL model also achieves a substantial reduction in training time compared to competing approaches.

References

[1] J. Li, M. Jing, K. Lu, L. Zhu, Y. Yang, and Z. Huang, "From zero-shot learning to cold-start recommendation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4189–4196.

[2] S. Sedhain, A. Menon, S. Sanner, L. Xie, and D. Braziunas, "Low-rank linear cold-start recommendation from social data," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, 2017, pp. 1502–1508.

[3] S. Jin, Y. Zhang, X. Li, and M. Lu, "Heterogeneous graph convolutional network for E-commerce product recommendation with adaptive denoising training," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3259–3268, Feb. 2024.

[4] H. Li, L. Sang, Y. Zhang, X. Zhang, and Y. Zhang, "CETN: Contrast-enhanced through network for click-through rate prediction," *ACM Trans. Inf. Syst.*, vol. 43, no. 1, p. 4, 2024.

[5] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2008, pp. 426–434.

[6] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.

[7] S. Ul-Durar, U. Awan, A. Varma, S. Memon, and A.-L. Mention, "Integrating knowledge management and orientation dynamics for organization transition from eco-innovation to circular economy," *J. Knowl. Manag.*, vol. 27, no. 8, pp. 2217–2248, 2023.

[8] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.
[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, 2001, pp. 285–295.
[10] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[11] X. Ning and G. Karypis, "Slim: Sparse linear methods for top-n recommender systems," in *Proc. IEEE 11th Int. Conf. Data Min.*, 2011, pp. 497–506.
[12] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proc. Web Conf.*, 2020, pp. 2704–2710.
[13] S. Liu, I. Ounis, C. Macdonald, and Z. Meng, "A heterogeneous graph neural model for cold-start recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 2029–2032.
[14] L. Sang, Y. Wang, Y. Zhang, Y. Zhang, and X. Wu, "Intent-guided heterogeneous graph contrastive learning for recommendation," 2024, *arXiv:2407.17234*.
[15] B. Xie, X. Ma, J. Wu, J. Yang, and H. Fan, "Knowledge graph enhanced heterogeneous graph neural network for fake news detection," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2826–2837, Feb. 2024.
[16] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 165–174.
[17] Y. Wang, L. Sang, Y. Zhang, and Y. Zhang, "Generative-contrastive heterogeneous graph neural network," 2024, *arXiv:2404.02810*.
[18] X. Wang et al., "Heterogeneous graph attention network," in *Proc. World Wide Web Conf.*, 2019, pp. 2022–2032.
[19] Y. Xu, Z. Qiu, H. Gao, X. Zhao, L. Wang, and R. Li, "Heterogeneous data-driven failure diagnosis for microservice-based industrial clouds toward consumer digital ecosystems," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2027–2037, Feb. 2024.
[20] Z. Lin, C. Tian, Y. Hou, and W. X. Zhao, "Improving graph collaborative filtering with neighborhood-enriched contrastive learning," in *Proc. ACM Web Conf.*, 2022, pp. 2320–2329.
[21] J. Wu et al., "Self-supervised graph learning for recommendation," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 726–735.
[22] L. Xia, C. Huang, and C. Zhang, "Self-supervised hypergraph transformer for recommender systems," in *Proc. 28th ACM SIGKDD Conf. Knowl. Disc. Data Min.*, 2022, pp. 2100–2109.
[23] B. Xu, J. Wang, Z. Zhao, H. Lin, and F. Xia, "Unsupervised anomaly detection on attributed networks with graph contrastive learning for consumer electronics security," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 4062–4072, Feb. 2024.
[24] H. Badawi, O. A. Arqub, and N. Shawagfeh, "Well-posedness and numerical simulations employing Legendre-shifted spectral approach for Caputo–Fabrizio fractional stochastic integrodifferential equations," *Int. J. Modern Phys. C*, vol. 34, no. 6, 2023, Art. no. 2350070.
[25] U. Awan, R. Sroufe, and K. Bozan, "Designing value chains for industry 4.0 and a circular economy: A review of the literature," *Sustainability*, vol. 14, no. 12, p. 7084, 2022.
[26] X. Wang, N. Liu, H. Han, and C. Shi, "Self-supervised heterogeneous graph neural network with co-contrastive learning," in *Proc. 27th ACM SIGKDD Conf. Knowl. Disc. Data Min.*, 2021, pp. 1726–1736.
[27] X. Long et al., "Social recommendation with self-supervised metagraph informax network," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manag.*, 2021, pp. 1160–1169.
[28] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, and R. Luo, "Heterogeneous graph contrastive learning for recommendation," in *Proc. 16th ACM Int. Conf. Web Search Data Min.*, 2023, pp. 544–552.
[29] B. Xu, J. Wang, Z. Zhao, H. Lin, and F. Xia, "Unsupervised anomaly detection on attributed networks with graph contrastive learning for consumer electronics security," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 4062–4072, Feb. 2024.
[30] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, and Q. V. H. Nguyen, "Are graph augmentations necessary? simple graph contrastive learning for recommendation," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 1294–1303.
[31] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 639–648.
[32] X. Yu et al., "A privacy-preserving cross-domain recommendation algorithm for industrial IoT devices," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 227–237, Feb. 2024.
[33] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–16.
[34] X. Wang and M. Zhang, "How powerful are spectral graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 23341–23362.
[35] P. Lv et al., "DSMN: An improved recommendation model for capturing the multiplicity and dynamics of consumer interests," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1236–1245, Feb. 2024.
[36] A. C. M. Fong, B. Zhou, S. C. Hui, G. Y. Hong, and T. A. Do, "Web content recommender system based on consumer behavior modeling," *IEEE Trans. Consum. Electron.*, vol. 57, no. 2, pp. 962–969, May 2011.
[37] X. Cai, C. Huang, L. Xia, and X. Ren, "LightGCL: Simple yet effective graph contrastive learning for recommendation," 2023, *arXiv:2302.08191*.
[38] H. Zhang et al., "Privacy-preserving federated heterogeneous graph learning via pseudo-metapath generation," *IEEE Trans. Consum. Electron.*, early access, Nov. 28, 2024, doi: 10.1109/TCE.2024.3503652.
[39] Y. Bi, L. Song, M. Yao, Z. Wu, J. Wang, and J. Xiao, "A heterogeneous information network based cross domain insurance recommendation system for cold start users," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 2211–2220.
[40] I. Fernández-Tobías, M. Braunhofer, M. Elahi, F. Ricci, and I. Cantador, "Alleviating the new user problem in collaborative filtering by exploiting personality information," *User Model. User-Adapted Interact.*, vol. 26, pp. 221–255, Feb. 2016.
[41] Z. E. Abo-Hammour, O. Alsmadi, S. Momani, and O. Abu Arqub, "A genetic algorithm approach for prediction of linear dynamical systems," *Math. Problems Eng.*, vol. 2013, no. 1, 2013, Art. no. 831657.
[42] Z. Zhu, X. Fan, X. Chu, and J. Bi, "HGCN: A heterogeneous graph convolutional network-based deep learning model toward collective classification," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2020, pp. 1161–1171.
[43] Y. Dong, N. V. Chawla, and A. Swami, "Metapath2Vec: Scalable representation learning for heterogeneous networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2017, pp. 135–144.
[44] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.
[45] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2012, *arXiv:1205.2618*.
[46] H. Wang, K. Zhou, X. Zhao, J. Wang, and J.-R. Wen, "Curriculum pre-training heterogeneous subgraph transformer for top-n recommendation," *ACM Trans. Inf. Syst.*, vol. 41, no. 1, pp. 1–28, 2023.
[47] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 995–1000.
[48] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Min.*, 2015, pp. 1235–1244.
[49] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.
[50] D. Zhang et al., "RecDCL: Dual contrastive learning for recommendation," 2024, *arXiv:2401.15635*.