



 PDF Download
3750725.pdf
17 December 2025
Total Citations: 2
Total Downloads: 440

 Latest updates: <https://dl.acm.org/doi/10.1145/3750725>

RESEARCH-ARTICLE

Bottlenecked Heterogeneous Graph Contrastive Learning for Robust Recommendation

LEI SANG, Anhui University, Hefei, Anhui, China

MAOHAO HUANG, Anhui University, Hefei, Anhui, China

YU WANG, Anhui University, Hefei, Anhui, China

YIWEN ZHANG, Anhui University, Hefei, Anhui, China

XINDONG WU, Hefei University of Technology, Hefei, Anhui, China

Published: 10 September 2025
Online AM: 24 July 2025
Accepted: 19 July 2025
Revised: 19 May 2025
Received: 28 September 2024

[Citation in BibTeX format](#)

Open Access Support provided by:

Hefei University of Technology

Anhui University

Bottlenecked Heterogeneous Graph Contrastive Learning for Robust Recommendation

LEI SANG, MAOHAO HUANG, YU WANG, and YIWEN ZHANG, School of Computer Science and Technology, Anhui University, Hefei, China
XINDONG WU, Hefei University of Technology, Hefei, China

In recommender systems, heterogeneous graph neural networks (HGNNs) have demonstrated remarkable efficacy due to their capacity to harness rich auxiliary information within heterogeneous information networks (HINs). However, existing HGNN-based recommendation faces severe noise cascading challenge. The presence of substantial data noise can adversely affect robustness of recommender, as the graph structures are susceptible to noise and even unnoticed malicious perturbations. Moreover, these noises can propagate and accumulate through connected nodes, potentially exerting a profound impact on target nodes within the graph structure. To tackle the noise challenges, we present a Bottlenecked Heterogeneous Graph Contrastive Learning (BHGCL), aiming to enhance the robustness of recommendation systems. BHGCL can first effectively separate fine-grained latent factors from complex self-supervision signals with a disentangled-based encoder, leveraging diverse semantic information across various meta-paths. Then, by employing the information bottleneck (IB) principle, BHGCL adaptively learns to reduce noise in augmented graphs. IB can capture the minimum sufficient information from the data features, which significantly improves system performance in environments with noisy data. Experimental findings from multiple real-world datasets reveal that our approach surpasses the latest advanced recommendation systems, verifying its effectiveness and robustness. To reproduce our work, we have open-sourced our code at <https://github.com/DuellingSword/BHGCL>.

CCS Concepts: • Information systems → Recommender systems;

Additional Key Words and Phrases: Heterogeneous Graph Neural Networks, Contrastive Learning, Disentangled Recommendation, Information Bottleneck

ACM Reference format:

Lei Sang, Maohao Huang, Yu Wang, Yiwen Zhang, and Xindong Wu. 2025. Bottlenecked Heterogeneous Graph Contrastive Learning for Robust Recommendation. *ACM Trans. Inf. Syst.* 43, 6, Article 162 (September 2025), 36 pages.

<https://doi.org/10.1145/3750725>

This work is supported by the National Natural Science Foundation of China (No. 62206002 and 62272001) and the Anhui Provincial Natural Science Foundation (220805QF195).

Authors' Contact Information: Lei Sang, School of Computer Science and Technology, Anhui University, Hefei, China; e-mail: sanglei@ahu.edu.cn; Maohao Huang, School of Computer Science and Technology, Anhui University, Hefei, China; e-mail: e23301222@stu.ahu.edu.cn; Yu Wang, School of Computer Science and Technology, Anhui University, Hefei, China; e-mail: wangyuahu@stu.ahu.edu.cn; Yiwen Zhang (corresponding author), School of Computer Science and Technology, Anhui University, Hefei, China; e-mail: zhangyiwen@ahu.edu.cn; Xindong Wu, Hefei University of Technology, Hefei, China; e-mail: xwu@hfut.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 1558-2868/2025/9-ART162

<https://doi.org/10.1145/3750725>

1 Introduction

In recent years, e-commerce and social media platforms have increasingly adopted recommender systems, significantly enhancing user experience and business value. Traditional recommender methods, such as **Collaborative Filtering (CF)** techniques, including approaches like matrix factorization [19], have achieved considerable success in practical applications. These methods primarily consider direct, single-hop connections in user-item interactions. Different from traditional methods, **Graph Neural Network (GNN)**-driven recommender systems can effectively model intricate, higher-order interactions, enabling a richer representation of user and item characteristics for enhanced recommendation performance [49, 50]. Despite the remarkable success, the majority of traditional recommender methods and current GNN-based models primarily rely on the user-item bipartite graph [49, 58], leading to sparse user-item interaction data. They fail to utilize the abundant heterogeneous data commonly found across real-world scenarios [75], which includes rich contextual information on both the user and item sides.

Heterogeneous Information Networks (HINs) provide powerful tools for recommendation systems, enabling flexible modeling of complex interactions between users and items [40]. Notably, meta-path, as composite relations within HINs, can capture the semantic associations between node pairs [33, 38]. Through meta-path, users can not only find items they have directly interacted with but also discover potential favorite items that are related to them via shared contextual information, effectively alleviating the challenges of limited data and new user/item scenarios in recommendation models [12]. For example, the connection between two movies can be discovered through meta-paths such as (1) “movie-user-user-movie” and (2) “movie-genre-movie.” These paths highlight latent aspects like (1) influence from social networks and (2) genre-based preferences [35]. Hence, we can find potentially interesting movies based on the intuition that the paths connecting two movies represent the movie relations of different semantics.

Several studies have explored the potential of GNNs to **Heterogeneous Graph Neural Networks (HGNNs)** in recommendation tasks and achieved impressive performance improvement by integrating auxiliary information from HINs [2, 32]. Despite these advances, challenges remain in learning robust and generalizable representations as user-item interactions in recommendation systems are often sparse and noisy [36]. Self-supervised learning approaches, like contrastive learning, have proven highly effective in enhancing representation quality and discriminative power by leveraging large amounts of unlabeled data and uncovering hidden structures within the data. Recently, the combination of contrastive learning with HGNN-based recommendation has gained attention as a leading approach for deriving generalizable representations from unlabeled data [31, 34]. In contrast to conventional CF approaches, these advanced methods apply the principles of alignment and uniformity [45] to maximize representation consistency across multiple views, thereby capturing more informative supervision signals. These advanced methods have been effectively utilized to strengthen the robustness and distinction of user and item embeddings in HGNN-based recommendation [1, 4]. For instance, HGCL [4] generates contrastive representations by combining supporting data from heterogeneous graphs and interaction patterns between users and items. These recommendation frameworks leverage the complex topological structures inherent in heterogeneous graphs, demonstrating promising potential to enhance user and item embedding learning across diverse recommendation scenarios.

However, recommendation systems built on HGNNs still suffer from heterogeneous noise introduced by diverse relationships in HINs. For example, in practical recommendation environments, implicit feedback frequently contains noisy signals, such as accidental clicks or interactions that do not reflect genuine user interests. Moreover, malicious users might fabricate fake profiles, either by linking to specific target users or by associating with fake accounts. We argue that such

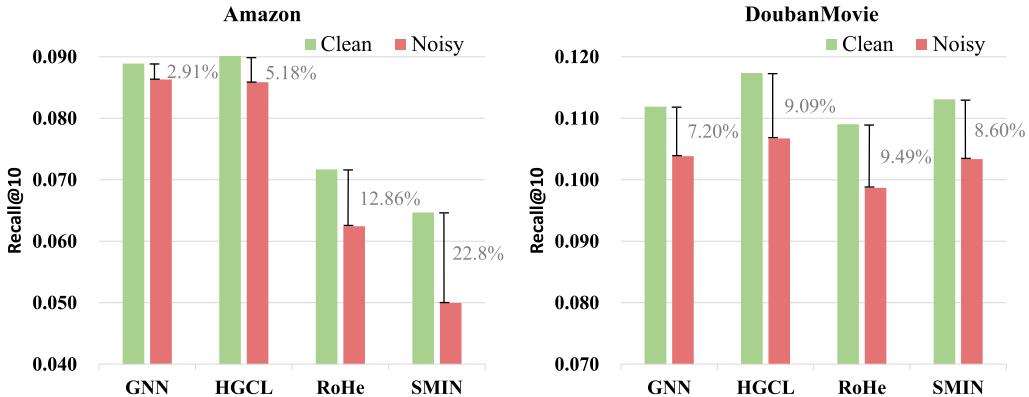


Fig. 1. Performance comparison of GNN-based model (utilizing user–item information) and other mainstream HGNN-based models (HGCL, RoHe, SMIN) on Amazon and DoubanMovie datasets under clean and noisy conditions. Results show that HGNN-based methods are more vulnerable to heterogeneous noise and exhibit weaker robustness.

noise-based attacks may be more detrimental to the training of HGNNs than to conventional neural networks. The reason lies in HGNNs’ information propagation process, which can amplify the impact of tampered nodes, causing cascading effects that influence a significant portion of the graph. As depicted in Figure 1, the GNN-based approach exhibits performance decreases of 2.91% and 7.20% on the Amazon and DoubanMovie datasets respectively when exposed to noise, whereas HGNN-based methods experience average performance drops of 13.61% on Amazon and 9.06% on DoubanMovie. This obvious difference in performance degradation reveals a critical tradeoff: while HGNN-based recommendation models achieve notable improvements over GNN-based approaches under clean conditions, they exhibit heightened sensitivity and pronounced vulnerability to noisy environments. Specifically, the iterative propagation operations in graph-based recommendations can cause cascading and cumulative effects of noise, severely undermining the robustness of recommendation systems and significantly degrading recommendation quality. In HINs, the introduction of meta-paths exacerbates this issue, as noise can propagate more extensively, amplifying and spreading it to a broader and more remote area of the graph.

In light of the challenges outlined above, we introduce a **Bottlenecked Heterogeneous Graph Contrastive Learning (BHGCL)** framework designed to bolster the robustness of recommendation. BHGCL initially disentangles similar-type interactions to capture fine-grained preferences and nuanced characteristics, which serves as a preliminary step in noise reduction by distinguishing meaningful patterns from potential noise in the data. It then performs **Cross-View Contrastive Learning (CVCL)** between these noise-filtered representations and embeddings learned from various meta-paths, each representing a specific latent intent, to fully assimilate heterogeneous side information. To address the pressing issue of noise within recommendation, we introduce a novel **Information Bottleneck (IB)**-based graph augmentation scheme. IB framework extracts only the **minimum sufficient information** from the underlying data characteristics, which significantly improving system performance in environments with noisy data. This approach synergizes with the disentanglement process to effectively identify and remove noisy interactions while accentuating informative structures. Consequently, BHGCL substantially strengthens the model’s resilience to data fluctuations and inconsistencies, maintaining strong performance and dependability in noisy, real-world recommendation environments. The contributions of this work are summarized below:

- We devise a heterogeneous graph disentangled contrastive learning framework that captures fine-grained user intents and leverages heterogeneous side information to enhance recommendation performance.
- We devise an IB-based graph augmentation scheme to ensure each view captures the minimum sufficient representation necessary for robust recommendation.
- We perform an exhaustive evaluation of BHGCL on multiple real-world datasets, with experimental evidence showing substantial improvements in accuracy and robustness over the most advanced current methods.

2 Preliminaries

This section elaborates on the essential concepts required to comprehend the framework and methodology applied in this work, including Setting of HIN, IB, Problem Formulation, and Notations.

2.1 Setting of HIN

HIN is a complex information structure, integrating diverse entity types and multifaceted relationships. Our recommendation framework is built upon key HIN-related concepts, which we formally characterize as follows:

HIN. We define a HIN as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}, \phi, \psi)$, where \mathcal{V} represents the set of nodes, and \mathcal{E} denotes the set of edges connecting these nodes. Unlike homogeneous information networks, where all nodes and edges are of the same type, a HIN consists of multiple types of nodes and edges. The diversity of both nodes and edges in a HIN is captured by the mappings $\phi : \mathcal{V} \rightarrow \mathcal{A}$ and $\psi : \mathcal{E} \rightarrow \mathcal{R}$, which assign each node $v \in \mathcal{V}$ to a specific type in the node type set \mathcal{A} , and each edge $e \in \mathcal{E}$ to its corresponding type in the edge type set \mathcal{R} . The sets \mathcal{A} and \mathcal{R} denote collections of distinct node and edge types, respectively. To ensure the network's heterogeneity, we impose the condition that $|\mathcal{A}| + |\mathcal{R}| > 2$.

Meta-Path and Meta-Path-Based Neighbors. In a HIN, a sequence of alternating node and edge types is referred to as a meta-path. This sequence, denoted as $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$, or simply $A_1 A_2 \dots A_{l+1}$, captures the connections between nodes through a chain of relationships. The relations between node types A_1 and A_{l+1} are established through edge types R_1, R_2, \dots, R_l , where \circ signifies the composition of these interactions. For each node i within the network, its meta-path-based neighbors, represented by \mathcal{N}_i^Φ , consist of all nodes that can be reached by following the specified meta-path \mathcal{P} . Note that a node is always considered a neighbor of itself.

User–Item Interaction Modeling and Meta-Path-Based Subgraph. We can model the connections between users and items using a graph $\mathcal{G}_{ui} = \{\mathcal{V}_u, \mathcal{V}_i, \mathcal{E}_{ui}\}$, where users and items are represented by \mathcal{V}_u and \mathcal{V}_i , respectively. In \mathcal{G}_{ui} , each edge $(u, i) \in \mathcal{E}_{ui}$ corresponds to an interaction between user u and item i , representing the fact that user u has performed some action related to item i . By applying various meta-paths, we can generate Meta-path-based Subgraphs that capture homogeneous interactions involving users and items. Specifically, we can derive user-user and item-item interaction graphs, denoted as $\mathcal{G}_{uu} = \{\mathcal{V}_u, \mathcal{E}_{uu}\}$ and $\mathcal{G}_{ii} = \{\mathcal{V}_i, \mathcal{E}_{ii}\}$, respectively, from the original HIN. These graphs, \mathcal{G}_{uu} and \mathcal{G}_{ii} , are examples of Meta-path-based Subgraphs that simplify the heterogeneous graph into homogeneous interaction networks, focusing on specific types of relationships.

2.2 IB

The IB approach principle on reducing the complexity of input data by filtering out irrelevant information, leaving behind only what is crucial for predicting the target labels. In the context of a dataset X , which includes unprocessed graph structure and raw node features, the IB method

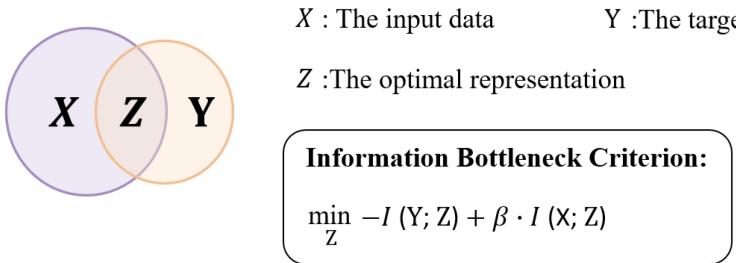


Fig. 2. A visual representation of the fundamental idea of the IB method. X represents the raw input data, which includes unprocessed and noise from the graph's topology and its node attributes. Y is the target label, and Z denotes the optimal representation that captures the *minimal sufficient information* from X to predict Y . The IB Criterion is used to find the representation Z that retains only the relevant information from X , while filtering out irrelevant noise for effective prediction of Y . Here, $I(\cdot; \cdot)$ denotes the mutual information, and β is a tradeoff parameter that controls the extent to which Z is compressed.

focuses on deriving an optimal representation Z that captures the *minimal sufficient information* required for predicting the target Y . As illustrated in Figure 2, X represents the input data in its original, unrefined state, which often includes noisy, redundant, or irrelevant information. This raw input, if not properly processed, can lead to overfitting or model instability. Y , on the other hand, is the target label (e.g., a class or a decision variable), and Z is the distilled representation that balances the tradeoff between compressing the input X and retaining the essential information necessary for the accurate prediction of Y . The core idea of the IB framework is to optimize that Z effectively captures the information needed to predict Y , while discarding irrelevant details from the unprocessed X . A larger β value encourages more compression of Z , potentially discarding some information about X that may not be useful for predicting Y . Conversely, a smaller β value results in a less compressed representation, possibly retaining more information from X , which may or may not be relevant for Y . The IB method focuses on ensuring that Z serves as a minimal sufficient statistic for Y , capturing all the relevant information for prediction while eliminating any superfluous data from X . This approach aids in mitigating overfitting and enhancing the model's robustness, particularly in challenging conditions such as noisy environments or when facing adversarial attacks.

2.3 Problem Formulation

We define $\mathcal{V}_u = \{u_1, u_2, \dots, u_M\}$ as the set of users and $\mathcal{V}_i = \{i_1, i_2, \dots, i_N\}$ as the set of items. The interaction matrix $Y \in \{0, 1\}^{M \times N}$ records whether user u_m has engaged with item i_n , where $y_{ui} = 1$ indicates an interaction, and $y_{ui} = 0$ represents no engagement. Furthermore, contextual data from the HIN \mathcal{G} , such as user-group and item-type relations, is also utilized. Our goal is to predict unobserved interactions within Y by learning disentangled representations for user and item derived from different aspects within the training set. A function $\hat{y}_{ui} = f(u, i | \Theta, \mathcal{G})$ is introduced to quantify the interaction potential between user u and item i , with Θ serving as the set of model parameters.

2.4 Notations

To facilitate the subsequent discussion of the model, we first outline the primary symbols used in this article. Table 1 provides an overview of these notations, which will be introduced step by step. The symbols are deliberately chosen to describe elements like input data, internal representations, and output labels, offering clarity and consistency in understanding the model's design and function.

Table 1. Notations and Explanations

Notation	Explanation
\mathcal{G}	Heterogeneous information network
\mathcal{V}, \mathcal{E}	Node and edge sets of graph \mathcal{G}
\mathcal{A}, \mathcal{R}	Categories of nodes and relationships
ϕ, ψ	Mapping functions for node and edge types
\mathcal{P}	Meta-path: $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$
\mathcal{N}_i^{Φ}	Meta-path-based neighbors of node i
\mathcal{G}_{ui}	Graph of user-item interactions
$\mathcal{G}_{uu}, \mathcal{G}_{ii}$	Meta-path-based subgraphs for users and items
$\mathcal{V}_u, \mathcal{V}_i$	Node sets for users and items, which are subsets of \mathcal{V}
\mathcal{E}_{ui}	Edges linking users and items in \mathcal{G}_{ui}
Y	Matrix representing user-item interactions, where $y_{ui} \in \{0, 1\}$
\hat{y}_{uv}	Predicted likelihood of interaction between user u and item i
Θ	Model parameters
c_u^k	Learnable global intent prototype for user u in IDE
$\mathbf{E}_u^D, \mathbf{E}_i^D$	Disentangled embeddings for users and items after IDE
$w_u^{\mathcal{P}_n}$	Weight for meta-path \mathcal{P}_n for user u
$\beta_u^{\mathcal{P}_n}$	Importance weight of meta-path \mathcal{P}_n for user u
\mathbf{W}, \mathbf{b}	Learnable parameters for the attention mechanism
\mathbf{q}	Semantic-level attention vector
$\mathbf{E}_u^M, \mathbf{E}_i^M$	Meta-path-based embeddings for users and items
\mathbf{E}	Combined representation matrix for users and items
l	Layer index in the IB Contrastive Learning module
$\omega_i^{(l)}, \omega_{e_{ui}}^{(l)}$	Masking parameters for node i and edge e_{ui} at layer l
$\rho_i^{(l)}, \rho_{e_{ui}}^{(l)}$	Binary mask values for node i and edge e_{ui} at layer l
$\mathcal{G}'_{UL}^{(l)}, \mathcal{G}'_{ED}^{(l)}$	Graphs from Node-Dropping and Edge-Dropping views of user-item network at layer l
$\mathbf{E}_{ND}^{(l)}, \mathbf{E}_{ED}^{(l)}$	Representations from Node-Dropping and Edge-Dropping views at layer l
$\tilde{\mathbf{E}}$	Augmented views of the user-item graph

IDE, Intention-guided Disentangled Encoder.

3 Methodology

Here, we offer a detailed overview of the BHGCL model, consisting of four key modules specifically designed to achieve the robustness of the recommendation system. (1) We first disentangle user-item interactions using the **Intention-Guided Disentangled Encoder (IDE)** to capture fine-grained user intents and item attributes, focusing on meaningful latent factors; (2) Next, we aggregate high-order relationships by integrating diverse meta-path information through the **Semantic-Guided Meta-Path Encoder (SME)**, which captures complex dependencies in heterogeneous graphs; (3) We then align and refine the embeddings from both encoders using CVCL, enhancing consistency and robustness; (4) Lastly, we apply an IB mechanism to eliminate noise and preserve only the most essential information, further improving the reliability of the recommendations. The complete framework is illustrated in Figure 3.

3.1 Intention-Guided Disentangled Encoder

The IDE is crucial for isolating fine-grained user intents and item attributes from noisy interactions. The approach directs the model toward identifying relevant latent factors, which in turn strengthens

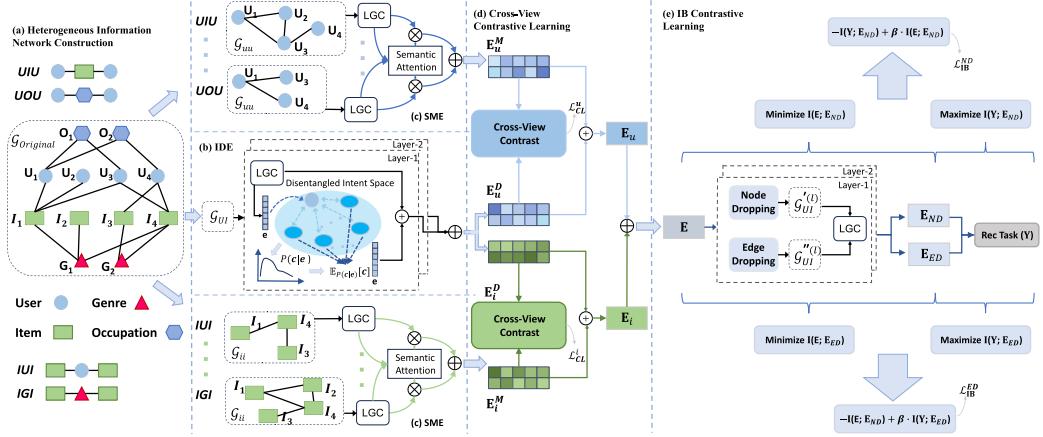


Fig. 3. The Framework of BHGCL. (a) Building a HIN consisting of diverse node and edge types. (b) Encoding the user-item graph using a two-layer encoder, referred to as IDE. (c) Aggregating nodes of the same type based on meta-path using the SME. (d) Using CVCL to align embeddings generated from (b) and (c). (e) IB adaptively learns to prune edges or nodes to refine graph structures, thereby effectively eliminating noise through an end-to-end process.

the embeddings' effectiveness. The learned intent prototypes demonstrate their effectiveness by capturing distinct and interpretable preference dimensions. For instance, in the e-commerce dataset, certain prototypes strongly align with category-specific preferences (e.g., electronics, fashion), while others capture temporal patterns like seasonal shopping behaviors [48]. Additionally, some prototypes model collaborative influence patterns, reflecting shared preferences within user groups. This multi-dimensional intent modeling enables our framework to provide more interpretable recommendations through explicit intent modeling, better handle cold-start scenarios by leveraging learned intent patterns, and capture fine-grained semantic preferences via prototype-specific predictions [23]. These findings extend recent work on disentangled representation learning to capture more nuanced patterns in recommendation scenarios.

The initial user and item embedding matrices serve as the inputs for the initial layer, denoted as $E^{(0)}$. These matrices are processed to generate disentangled embeddings E_u^D and E_i^D . Building on the streamlined and powerful architecture of the lightweight GCN [10], we apply a message propagation framework that iteratively updates both user and item representations. The embeddings are refined by aggregating information from their respective neighbors, leading to enhanced performance, as demonstrated in the following formulation:

$$\begin{aligned} h_u^{(l+1)} &= \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|} \sqrt{|\mathcal{N}_i|}} e_i^{(l)}, \\ h_i^{(l+1)} &= \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|} \sqrt{|\mathcal{N}_u|}} e_u^{(l)}. \end{aligned} \quad (1)$$

In this formulation, this processing method can be formulated using the **Light Graph Convolution (LGC)** paradigm as follows: \mathcal{N}_u and \mathcal{N}_i represent the neighboring nodes of u and i within the graph G_{ui} . The embeddings $e_u^{(l)}$ and $e_i^{(l)}$, extracted from their respective embedding matrices $E_u^{(l)}$ and $E_i^{(l)}$, are row vectors of dimensionality d at layer l . This process is consistent with the LGC

methodology, as outlined in the following equations:

$$\begin{aligned}\mathbf{H}_u^{(l+1)} &= \text{LGC}(\mathbf{E}_u^{(l)}, \mathcal{G}_{ui}), \\ \mathbf{H}_i^{(l+1)} &= \text{LGC}(\mathbf{E}_i^{(l)}, \mathcal{G}_{ui}).\end{aligned}\quad (2)$$

For brevity, we illustrate the following method on the user side without loss of generality. The embeddings \mathbf{E}_i^D can be obtained in a similar manner on the item side. Following the approach of DCCF [28], within our disentangled-based encoder, we capture user preferences through $\mathbb{E}_{P(c_u|\mathbf{e}_u^{(l)})}[c_u]$. We introduce K global intent prototypes for users, denoted as $\{c_u^k \in \mathbb{R}^d\}_{k=1}^K$, which are learnable intent embeddings. By leveraging these embeddings, we design the following method to generate user intent-aware representations. Here, $\mathbf{r}_u^{(l)}$ represents the row vector corresponding to user u within the user intent-aware embedding matrix $\mathbf{R}_u^{(l)}$.

$$\begin{aligned}\mathbf{r}_u^{(l)} &= \mathbb{E}_{P(c_u|\mathbf{e}_u^{(l)})}[c_u] \\ &= \sum_{k=1}^K c_u^k P(c_u^k|\mathbf{e}_u^{(l)}) \\ &= \sum_{k=1}^K c_u^k \frac{\exp(\mathbf{e}_u^{(l)\top} c_u^k)}{\sum_{k'}^K \exp(\mathbf{e}_u^{(l)\top} c_u^{k'})}, u \in \mathcal{V}_u.\end{aligned}\quad (3)$$

Next, we update the user embedding matrix using intent-aware representations and applying residual connections [3, 60] from layer l to layer $l + 1$. We formalize this process through the following mathematical formulation:

$$\mathbf{E}_u^{(l+1)} = \mathbf{E}_u^{(l)} + \mathbf{H}_u^{(l)} + \mathbf{R}_u^{(l)}. \quad (4)$$

By stacking L layers, the aggregation of user and item disentangled embeddings across different layers is achieved through the following formulation:

$$\mathbf{E}_u^D = \sum_{l=0}^L \mathbf{E}_u^{(l)} \quad \text{and} \quad \mathbf{E}_i^D = \sum_{l=0}^L \mathbf{E}_i^{(l)}. \quad (5)$$

3.2 Semantic-Guided Meta-Path Encoder

We utilize the SME to aggregate meta-path-based information, capturing both structural and semantic relationships within heterogeneous graphs. This approach allows the model to effectively integrate diverse contextual dependencies for more accurate recommendations. Different meta-paths naturally capture distinct semantic signals in our framework. For example, **User–Item–User (UIU)** paths reveal preference similarities between users with shared interactions, while Item–Category–Item paths uncover category-level semantic patterns. Our experiments demonstrate that users connected through UIU paths tend to share similar intent prototype distributions, validating the semantic consistency of our model. More complex paths like User–Item–Category–Item provide multi-hop semantic verification by connecting users through both direct interactions and categorical similarities. The attention mechanism ($w_u^{\mathcal{P}_n}$ and $\beta_u^{\mathcal{P}_n}$) automatically weighs these semantic signals, offering interpretable evidence for the model’s recommendation decisions through the learned attention weights.

To explain this process, we concentrate on the user’s perspective. Illustrated in Figure 3(a), the meta-path User–Occupation–User, U_1 has U_1 (itself), U_3 , and U_4 as its meta-path-based neighbors. Given a user node u ($u \in \mathcal{V}_u$) and M user-side meta-paths $\{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_M\}$, the meta-path-based neighbors $N_u^{\mathcal{P}_n}$ can be identified for each meta-path \mathcal{P}_n . Different paths capture unique semantic

relationships, and the corresponding information is processed using a specialized GCN [18] for every path:

$$\mathbf{h}_u^{\mathcal{P}_n} = \sum_{u' \in \mathcal{N}_u^{\mathcal{P}_n}} \frac{1}{\sqrt{|\mathcal{N}_u^{\mathcal{P}_n}|} \sqrt{|\mathcal{N}_{u'}^{\mathcal{P}_n}|}} e_{u'}^0, \quad (6)$$

where $e_{u'}^0 \in \mathbb{R}^d$ is a row vector in the initial user embedding matrix \mathbf{E}_u^0 , representing the embedding of user u' . To achieve this, we calculate a weight $w_u^{\mathcal{P}_n}$ for each meta-path \mathcal{P}_n . This is done using a specialized attention mechanism that computes the relevance of each meta-path as follows:

$$w_u^{\mathcal{P}_n} = \frac{1}{|\mathcal{V}_u|} \sum_{u' \in \mathcal{V}_u} \mathbf{q}_u^\top \cdot \tanh(\mathbf{W}_u \mathbf{h}_{u'}^{\mathcal{P}_n} + \mathbf{b}_u). \quad (7)$$

Here, $w_u^{\mathcal{P}_n}$ denotes the unnormalized importance of meta-path \mathcal{P}_n for user u , which is subsequently processed by a softmax function, yielding the normalized coefficient $\beta_u^{\mathcal{P}_n}$:

$$\beta_u^{\mathcal{P}_n} = \frac{\exp(w_{\mathcal{P}_n})}{\sum_{n=1}^M \exp(w_{\mathcal{P}_n})}, \quad (8)$$

where $\mathbf{W}_u \in \mathbb{R}^{d \times d}$ denotes the transformation matrix, $\mathbf{b}_u \in \mathbb{R}^{d \times 1}$ serves as the bias component, while \mathbf{q}_u encodes the semantic-level representation for the user. Here, \mathbf{q}_u is a learnable semantic-level attention vector that is shared across all meta-paths to ensure meaningful comparison between different semantic spaces. This vector is trained end-to-end with the model to capture the relative importance of different semantic aspects. By applying semantic attention, we compute the representation rich in diverse semantic information through a weighted aggregation:

$$\mathbf{e}_u^M = \sum_{n=1}^M \beta_u^{\mathcal{P}_n} \cdot h_u^{\mathcal{P}_n}, \quad (9)$$

where \mathbf{e}_u^M is a row vector from the user embedding matrix \mathbf{E}_u^M under the meta-path-based view. Similarly, the item embedding matrix \mathbf{E}_i^M can be obtained using the same method.

3.3 CVCL

This section introduces CVCL, aimed at improving the robustness of user and item representations by aligning embeddings generated from different perspectives. CVCL aims to reconcile the distinct representations obtained from disentangled and meta-path-based encoders. By contrasting these views, our approach captures both the fine-grained intents of users and the broader contextual relationships, ensuring consistency and robustness across diverse interaction scenarios.

With the help of our Disentangled-Based Homogeneous Encoder generating \mathbf{E}_u^D (combining \mathbf{E}_u^D and \mathbf{E}_i^D) and the Meta-path-based Heterogeneous Encoder generating \mathbf{E}_u^M (combining \mathbf{E}_u^M and \mathbf{E}_i^M), and drawing inspiration from the recent developments in contrastive self-supervised methodologies applied to recommendation systems [58, 60], we strive to further refine the process of capturing user and item characteristics to deliver higher accuracy. This is realized using a contrastive objective grounded in InfoNCE, which is applied to two distinct views of the representations. Under the user's perspective, the cross-view contrastive objective can be formulated as:

$$\mathcal{L}_{\text{CL}}^u = \sum_{u \in \mathcal{V}_u} -\log \frac{\exp(s(\mathbf{e}_u^D, \mathbf{e}_u^M)/\tau)}{\sum_{u' \in \mathcal{V}_u} \exp(s(\mathbf{e}_u^D, \mathbf{e}_{u'}^M)/\tau)}. \quad (10)$$

In this context, $\mathbf{e}_u^D, \mathbf{e}_u^M \in \mathbb{R}^d$ represent the user embedding vectors extracted from the matrices \mathbf{E}_u^D and \mathbf{E}_u^M , respectively. The vectors are compared using cosine similarity, implemented via the function

$s(\cdot)$. The parameter τ acts as a temperature coefficient, adjusting the distribution smoothness of the similarity scores to better align the representations generated from the disentangled and heterogeneous views. Similarly, the InfoNCE loss $\mathcal{L}_{\text{CL}}^i$ can be derived under the item's perspective. Finally, the total contrastive loss is $\mathcal{L}_{\text{CL}} = \mathcal{L}_{\text{CL}}^u + \mathcal{L}_{\text{CL}}^i$.

The final enriched disentangled representation of user and item, incorporating information from the HIN, can be expressed as follows:

$$\mathbf{E}_u = 0.5 * \mathbf{E}_u^D + 0.5 * \mathbf{E}_u^M, \quad \mathbf{E}_i = 0.5 * \mathbf{E}_i^D + 0.5 * \mathbf{E}_i^M. \quad (11)$$

3.4 IB Contrastive Learning

While directly using the embeddings \mathbf{E} (combining \mathbf{E}_u and \mathbf{E}_i) that have sufficiently incorporated the disentangled potential intentions and heterogeneous information can be applied to complete the recommendation task, we have observed that relying solely on \mathbf{E} is insufficient. This is because a large amount of harmful or useless noise exists in HIN, along with the influence of many popular nodes. To address this issue, inspired by CGI [55], we adopt the IB principle to adaptively learn two augmented graphs. Their information is represented in a compressed form through IB-Regularized Optimization, which efficiently and adaptively reduces the redundant noise in the augmented graphs.

Compared to traditional augmentation and regularization methods, our IB-based approach offers several key theoretical advantages. While random augmentation methods (e.g., SGL [58], NCL [20]) use fixed dropout ratios and feature masking approaches (e.g., HCCF [60], MHCN [67]) rely on pre-defined patterns, the IB framework provides principled bounds on information compression through $I(\mathbf{E}' ; \tilde{\mathbf{E}}) \leq I(\mathbf{E} ; \tilde{\mathbf{E}}) - I(\mathbf{Y} ; \tilde{\mathbf{E}})$. This theoretical foundation guarantees the preservation of task-relevant information while removing noise, offering mathematical justification for our augmentation strategy. Furthermore, our approach realizes data-dependent augmentation through learnable parameters: $\omega_i^{(l)} = \text{MLP}(\mathbf{h}_i^{(l)})$ and $\omega_{e_{ui}}^{(l)} = \text{MLP}([\mathbf{h}_u^{(l)} ; \mathbf{h}_i^{(l)}])$. This adaptive mechanism elegantly addresses heterogeneous graph structures and automatically adjusts to diverse structural patterns, in contrast to traditional regularization techniques (e.g., L1/L2, dropout) which apply uniform regularization strength across all components and may lead to over-smoothing in complex heterogeneous relations.

3.4.1 Adaptive Multi-View Augmentation on Graph Structure. In the multi-view graph augmentation section, our objective is to adaptively enhance the user-item graph through a seamless and automated way. So we first adopt **Multi-Layer Perceptrons (MLPs)** to control the parameters $\omega_i^{(l)}$ and $\omega_{e_{ui}}^{(l)}$, which determine whether to mask the node i and the edge e_{ui} ($(u, i) \in \mathcal{E}_{ui}$), respectively. The vectors \mathbf{e}_u and \mathbf{e}_i are row vectors from the matrix \mathbf{E} , which fully incorporates decoupled latent intentions and heterogeneous information. The masking mechanism is formulated as:

$$\omega_i^{(l)} = \text{MLP}^{(l)}(\mathbf{e}_i), \quad \omega_{e_{ui}}^{(l)} = \text{MLP}^{(l)}([\mathbf{e}_u; \mathbf{e}_i]). \quad (12)$$

To efficiently enhance the comprehensive learning of multi-view structures, we adopt the reparameterization trick [16] to derive the mask ρ .

$$\rho = \sigma\left(\frac{\log \epsilon - \log(1 - \epsilon) + \omega}{\tau'}\right). \quad (13)$$

Here, ϵ is generated from a uniform distribution within the range $(0, 1)$, $\tau' \in \mathbb{R}^+$ serves as a temperature scaler that adjusts the smoothness of the mask generation process, and $\sigma(\cdot)$ denotes the sigmoid function.

Subsequently, we designed two distinct types of adaptive augmentation views for the graph structure:

– **Node-Dropping (ND) View:** We apply an adaptive node dropping technique at each layer to selectively diminish the influence of prominent nodes, resulting in the ND view in \mathcal{G}_{UI} . This operation is mathematically formulated as:

$$\mathcal{G}'^{(l)}_{UI} = \left\{ \{i \odot \rho_i^{(l)} \mid i \in \mathcal{V}\}, \mathcal{E}_{ui} \right\}, \quad (14)$$

where $\rho_i^{(l)} \in \{0, 1\}$ is sampled from a Bernoulli process governed by the parameter $\omega_i^{(l)}$, meaning $\rho_i^{(l)}$ follows the distribution $\text{Bern}(\omega_i^{(l)})$, indicating whether the node i is retained. Simply discarding sampled nodes and their edges severely disrupts the bipartite graph structure, hinders information aggregation, and destabilizes training. Therefore, we replace the selected nodes with their local subgraph representations while preserving their edges. We generate the local subgraph for node i through random walks and apply mean pooling to the sampled nodes.

– **Edge-Dropping (ED) View:** Analogous to the ND view, we generate the ED view through a learnable edge pruning technique:

$$\mathcal{G}''^{(l)}_{UI} = \{\mathcal{V}, \{e_{ui} \odot \rho_{ij}^{(l)} \mid e_{ui} \in \mathcal{E}_{ui}\}\}, \quad (15)$$

where $\rho_{e_{ui}}^{(l)} \in \{0, 1\}$ also follows $\rho_{e_{ui}}^{(l)} \sim \text{Bern}(\omega_{e_{ui}}^{(l)})$ and denotes whether the edge e_{ij} exists.

Next, we apply LGCs to derive user and item representations from these views:

$$\mathbf{E}_{ND}^{(l)} = \text{LGC}(\mathbf{E}_{ND}^{(l-1)}, \mathcal{G}'^{(l)}_{UI}), \quad \mathbf{E}_{ED}^{(l)} = \text{LGC}(\mathbf{E}_{ED}^{(l-1)}, \mathcal{G}''^{(l)}_{UI}). \quad (16)$$

In this case, the initial representations are given by $\mathbf{E}_{ND}^{(0)} = \mathbf{E}_{ED}^{(0)} = \mathbf{E}$. After stacking L LightGCN layers, we use a mean operation to form the final representations \mathbf{E}_{NF} and \mathbf{E}_{EF} . For brevity, we will use $\tilde{\mathbf{E}}$ to represent the augmented views, omitting the specific augmentation types *ND* and *ED* in subsequent notation.

3.4.2 IB-Regularized Optimization. Simply using the recommendation objective is insufficient to generate high-quality augmented views. Thus, we incorporate the IB principle. Unlike conventional contrastive learning, our method focuses on enhancing the divergence between the representations learned in the augmented view and the information in the original graph, while still retaining information pertinent to the recommendation task. This approach effectively removes redundant noise. The objective function, regularized by IB, takes the following form:

$$\mathcal{L}_{IB} = -I(\mathbf{Y}; \tilde{\mathbf{E}}) + \beta \cdot I(\mathbf{E}; \tilde{\mathbf{E}}), \quad (17)$$

where \mathcal{L}_{IB} can be used to derive \mathcal{L}_{IB}^{ND} and \mathcal{L}_{IB}^{ED} separately. Here, $\tilde{\mathbf{E}}$ denotes the representations of the augmented views, and \mathbf{Y} represents the downstream recommendation information. $I(\mathbf{Y}; \tilde{\mathbf{E}})$ refers to the **Bayesian Personalized Ranking (BPR)** [30] loss calculated from the augmented view representations as the reconstruction term, and $I(\mathbf{E}; \tilde{\mathbf{E}})$ quantifies the mutual information linking the original embeddings with their augmented counterparts, acting as the minimality term. β trades off the reconstruction term and the minimization term, where a larger value indicates a more compact compression representation.

According to [43], reducing the InfoNCE loss [8] is directly related to the increase in the lower bound of mutual information that we are attempting to approximate. To approximate the mutual information mentioned earlier, we utilize the negative InfoNCE loss. Below is the method for calculating the mutual information on the user side:

$$I(\mathbf{E}_u; \tilde{\mathbf{E}}_u) = \sum_{u \in \mathcal{V}_u} \log \frac{\exp(s(\mathbf{e}_u, \tilde{\mathbf{e}}_u)/\tau'')}{\sum_{u' \in \mathcal{V}_u} \exp(s(\mathbf{e}_u, \tilde{\mathbf{e}}_{u'})/\tau'')}, \quad (18)$$

In this scenario, $s(\cdot)$ represents the cosine similarity function used to compare two vectors. The hyper-parameter τ'' modulates the temperature, influencing how sensitive the similarity scores are during mutual information estimation. On the item side, mutual information $I(E_i; \tilde{E}_i)$ is calculated similarly, and the total mutual information is the sum of both user and item contributions: $I(E; \tilde{E}) = I(E_u; \tilde{E}_u) + I(E_i; \tilde{E}_i)$.

Notably, τ'' fine-tunes the sensitivity of cosine similarity in the context of mutual information, distinguishing it from the parameters τ and τ' mentioned earlier. The parameter τ , as seen in Equation (10), adjusts the distribution smoothness of similarity scores to better align the representations. Meanwhile, τ' , as shown in Equation (13), controls the smoothness of the mask generation process. On the other hand, τ'' directly impacts the precision of mutual information estimation.

3.4.3 Theoretical Analysis of IB-Regularized Optimization. Formally, Let E denote the ensemble of user and item representations before augmentation (pre-augmentation), \tilde{E} denote the representations of these augmented views (post-augmentation), E' represent harmful noise or redundant and useless information before augmentation, and Y denote downstream recommendation information.

Assuming that E' is independent of Y , an upper bound for the mutual information $I(E'; \tilde{E})$ can be expressed as the difference between two mutual information terms, specifically $I(E; \tilde{E})$ and $I(Y; \tilde{E})$:

$$I(E'; \tilde{E}) \leq I(E; \tilde{E}) - I(Y; \tilde{E}). \quad (19)$$

PROOF. Based on the proof in [55] and the Markov chain method in [11], we assume that E is determined by Y and E' . Then, we can establish the Markov chain $(Y, E') \rightarrow E \rightarrow \tilde{E}$. This Markov chain describes a probabilistic dependence such that, given E , \tilde{E} is conditionally independent of (Y, E') . Combining this with the Data Processing Inequality, we arrive at the subsequent inequality:

$$I(E; \tilde{E}) \geq I((Y, E'); \tilde{E}). \quad (20)$$

The chain rule for mutual information states that for random variables A , B , and C , the following relationship holds:

$$I((A, B); C) = I(B; C) + I(A; C|B). \quad (21)$$

In Equation (21), let $A = (Y, E')$, $B = E'$, and $C = \tilde{E}$, then we have:

$$I((Y, E'); \tilde{E}) = I(E'; \tilde{E}) + I(Y; \tilde{E}|E'). \quad (22)$$

Next, leveraging the definition of conditional mutual information and the chain rule for conditional entropy, $I(Y; \tilde{E}|E')$ can be decomposed as follows, where $H(\cdot)$ denotes entropy, used to measure the uncertainty inherent in a random variable.

$$\begin{aligned} I(Y; \tilde{E}|E') &= H(Y, E') - H(E') \\ &\quad + H(\tilde{E}, E') - H(Y, \tilde{E}, E') \\ &= H(Y|E') - H(Y|E', \tilde{E}). \end{aligned} \quad (23)$$

Substituting Equation (23) in Equation (22) yields the expression for $I((Y, E'); \tilde{E})$:

$$I((Y, E'); \tilde{E}) = I(E'; \tilde{E}) + H(Y|E') - H(Y|E', \tilde{E}). \quad (24)$$

Considering that E' and Y are uncorrelated, we can deduce that $H(Y|E') = H(Y)$. Additionally, since more conditional variables provide more information, the residual uncertainty of Y (conditional entropy) will decrease or remain the same. Therefore, we have $H(Y|E', \tilde{E}) \leq H(Y|\tilde{E})$. Next, we first

substitute Equation (24) into Equation (20), and then perform the following simplification steps:

$$\begin{aligned}
 I(\mathbf{E}; \tilde{\mathbf{E}}) &\geq I((\mathbf{Y}, \mathbf{E}'); \tilde{\mathbf{E}}) \\
 &= I(\mathbf{E}'; \tilde{\mathbf{E}}) + H(\mathbf{Y}|\mathbf{E}') - H(\mathbf{Y}|\mathbf{E}', \tilde{\mathbf{E}}) \\
 &= I(\mathbf{E}'; \tilde{\mathbf{E}}) + H(\mathbf{Y}) - H(\mathbf{Y}|\tilde{\mathbf{E}}) \\
 &= I(\mathbf{E}'; \tilde{\mathbf{E}}) + I(\mathbf{Y}; \tilde{\mathbf{E}}).
 \end{aligned} \tag{25}$$

Therefore, we have $I(\mathbf{E}'; \tilde{\mathbf{E}}) \leq I(\mathbf{E}; \tilde{\mathbf{E}}) - I(\mathbf{Y}; \tilde{\mathbf{E}})$ proved, with $I(\mathbf{Y}; \tilde{\mathbf{E}})$ aligning with \mathcal{L}_{BPR} from Equation (17), and $I(\mathbf{E}; \tilde{\mathbf{E}})$ corresponding to Equation (18).

In summary, through the *Proof* provided above for Equation (19), we can see that optimizing Equation (17) enables BHGCL to promote noise invariance during augmentation by effectively compressing information.

By optimizing the IB objective function (17), we can reduce the mutual information $I(\mathbf{E}; \tilde{\mathbf{E}})$ between the augmented feature $\tilde{\mathbf{E}}$ and the original feature \mathbf{E} , while simultaneously preserving the task-relevant information $I(\mathbf{Y}; \tilde{\mathbf{E}})$. The process ensures that the upper limit on the right side of inequality (19) is tightened, leading to the compression of the mutual information on the left-hand side, $I(\mathbf{E}'; \tilde{\mathbf{E}})$. Consequently, this helps in reducing noise and preserving essential information in the augmented feature representation, thereby improving the model's robustness to noise.

Furthermore, due to the guarantees provided by the rigorous IB theory, IB contrastive learning can minimize the intake of irrelevant interaction information while ensuring that no information essential for the recommendation task's predictive capability is lost.

3.5 Optimization Objectives of BHGCL

Our BHGCL model leverages fused embeddings \mathbf{E}_u and \mathbf{E}_i to forecast user–item interactions using a dot-product: $\hat{y}_{u,i} = \mathbf{e}_u^\top \mathbf{e}_i$, where \mathbf{e}_u corresponds to the embedding of user u and \mathbf{e}_i corresponds to the embedding of item i , both derived from \mathbf{E} . During training, each sample includes a user u , a positive item i^+ (previously interacted with by the user), and a negative item i^- (no prior interaction). BPR [30] assumes that interactions which have been observed are more representative of user preferences than those which have not been observed. Consequently, these observed interactions should receive higher predicted values. The optimization goal is defined as follows:

$$\mathcal{L}_{BPR} = \sum_{(u, i^+, i^-) \in O} -\ln(\text{softplus}(\hat{y}_{u,i^+} - \hat{y}_{u,i^-})) + \lambda_1 \|\Theta\|_2^2. \tag{26}$$

In this formulation, Θ refers to the set of parameters defining the model. The overall training loss integrates several components: \mathcal{L}_{IB}^{ND} and \mathcal{L}_{IB}^{ED} from the ND and ED views, respectively, as well as \mathcal{L}_{CL} from CVCL. These components are combined within the BPR [30] loss framework and can be expressed as follows. The weights of the L_2 regularization and the cross-domain contrastive loss are controlled by the hyperparameters λ_1 and λ_2 , respectively.

$$\mathcal{L} = \mathcal{L}_{BPR} + \mathcal{L}_{IB}^{ND} + \mathcal{L}_{IB}^{ED} + \lambda_2 \cdot \mathcal{L}_{CL}. \tag{27}$$

3.6 Complexity Analysis of BHGCL

We now evaluate the time complexity of the main components within our BHGCL, as outlined below:

- (i) The time complexity of the IDE's propagation process is given by $O(L \times |\mathcal{E}_{ui}| \times d + L \times (|\mathcal{V}_u| + |\mathcal{V}_i|) \times K \times d)$, where the first term refers to the complexity of the LGC component, while the second term corresponds to the intent-aware disentanglement mechanism. The variables are defined as follows: L specifies the number of graph propagation layers, K refers to the

global intent prototypes, $|\mathcal{E}_{ui}|$ represents the edges in the user–item network, $|\mathcal{V}_u|$ and $|\mathcal{V}_i|$ enumerate the users and items, respectively, and d is the dimensionality of the embeddings.

- (ii) The aggregation phase of the SME introduces a computational complexity of $O(M \times (|\mathcal{E}_{uu}| + |\mathcal{V}_u| \times d))$, with M denoting the meta-paths processed on the user side. The item side follows the same computational demand.
- (iii) The adaptive graph augmentation procedure requires $O(L \times |\mathcal{E}_{ui}| + (|\mathcal{V}_u| + |\mathcal{V}_i|) \times d)$ when producing two distinct augmented views.
- (iv) The complexity of optimizing the objectives of BHGCL can be expressed as $O(3 \times (B + |\mathcal{V}_u|^2 + |\mathcal{V}_i|^2) \times d)$, where B defines the number of users/items per mini-batch.

4 Experiments

This section presents a comprehensive set of experiments conducted on various datasets to evaluate the performance of the BHGCL model. We structure the experiments to explore the following research questions:

- *RQ1*: How does BHGCL in recommendation performance compare against that of various existing baselines?
- *RQ2*: Do the various key components in BHGCL contribute to improving recommendation performance?
- *RQ3*: How does BHGCL perform in terms of robustness in noisy interactions?
- *RQ4*: Does BHGCL have the capability to mitigate popularity bias?

4.1 Datasets

We conducted a comprehensive evaluation of our BHGCL framework using five distinct and publicly accessible real-world datasets specifically designed for top-N recommendation: MovieLens¹, Amazon², Yelp³, DoubanBook [66], and DoubanMovie [69]. Table 2 provides a summary of the statistics for each dataset. To maintain high-quality data standards, we apply a filtering criterion where only interactions with a rating greater than 3 are retained, and both users and items are required to have a minimum of 5 interactions to be included.

The dataset is first divided by allocating 80% of each user’s historical interactions for training and keeping 20% for evaluation. Additionally, we randomly select 10% of the training examples to serve as a validation set for hyperparameter tuning. Positive samples are observed user–item interactions, while negative samples are created by pairing users with items they have not previously interacted with.

4.2 Baselines

A series of comparisons has been performed between BHGCL and multiple leading baseline models. The subsequent section provides an in-depth overview of these baselines.

- *HERec* [38]: Generates sequences of nodes through meta-path-guided random walks, and these sequences are then merged using fusion strategies and embedded in a matrix factorization model for recommendation.
- *NGCF+* [49]: We extend NGCF to utilize graph convolution for enhanced information exchange and node feature aggregation in complex heterogeneous graphs.

¹<https://grouplens.org/datasets/movielens/>.

²<http://jmcauley.ucsd.edu/data/amazon/>.

³<https://www.yelp.com/dataset/>.

Table 2. Statistical Details and Meta-Paths for the Datasets

Dataset (Sparsity)	#User	#Item	#Interaction	Meta-Paths
MovieLens (93.70%)	943	1,682	100,000	<i>UMU</i> <i>MUM, MGM</i>
Amazon (98.85%)	6,170	2,753	195,791	<i>UIU, IUI</i> <i>ICI, IBI, IVI</i>
Yelp (99.91%)	19,239	14,284	198,397	<i>UBU, BUB</i> <i>BCB^a, BCB^b</i>
DoubanBook (99.73%)	13,024	22,347	792,062	<i>UBU, UGU</i> <i>BUB, BAB</i>
DoubanMovie (99.37%)	13,367	12,677	1,068,278	<i>UMU, UGU, UUU</i> <i>MUM, MTM</i>

^aBusiness-City-Business.^bBusiness-Category-Business.

- *HAN* [51]: Utilizes an attention mechanism tailored for heterogeneous graphs to capture embeddings from user-based and item-based meta-paths within recommendation dataset.
- *DisenGCN* [22]: Leverages a dynamic routing strategy among neighbors to discover hidden factors and generate disentangled node embeddings.
- *HGT* [14]: Proposes a mutual attention-based message passing approach tailored for heterogeneous graphs, enhancing the embeddings of users/items, and their complex relationships to improve recommendation tasks. This approach completes mutual attention and recommendation tasks by utilizing soft paths connecting users and items in diverse network structures.
- *DGCF* [52]: Combines user intents by decomposing multiple latent factors to enhance recommendation performance. This method aims to capture different user interests and preferences to provide more personalized and accurate recommendations.
- *HeCo* [53]: A self-supervised method considering one-hop and high-order relationships, using multi-view learning embeddings for contrastive learning.
- *SMIN* [21]: A self-supervised recommendation model that leverages both social connections and **Knowledge Graphs (KGs)** to enrich user preference modeling and optimize CF outcomes.
- *RoHe* [74]: Uses the same framework as HAN, adopting an attention mechanism to defend against attacks.
- *HGCL* [4]: Based on recommendation-oriented heterogeneous graph contrastive learning, constructs two aligned views using heterogeneous information, and enhances user and item representations through self-supervised contrastive learning.
- *HGCL+*: We extend HGCL by incorporating the IB mechanism to enhance its robustness against noisy interactions.
- *CGI* [55]: Applies IB theory to refine graph structures by selectively pruning nodes and edges, facilitating multi-view representation learning for users and items, which helps address popularity bias and reduce interaction noise.
- *RecDCL* [71]: Proposes a dual contrastive learning framework for recommendation, combining **Batch-Wise Contrastive Learning (BCL)** and **Feature-Wise Contrastive Learning (FCL)** to eliminate redundant solutions and enhance the robustness of user and item representations.

Table 3. A Comparative Analysis of All Methods across Multiple Datasets, Using Recall@10 and NDCG@10 as Evaluation Metric

Dataset	MovieLens		Amazon		Yelp		DoubanBook		DoubanMovie	
Metrics	R@10	N@10								
(2018)HERec	0.1836	0.3455	0.0712	0.0605	0.0484	0.0356	0.1025	0.1166	0.1229	0.1752
(2019)NGCF+	0.1600	0.3007	0.0723	0.0584	0.0533	0.0407	0.0779	0.0916	0.1211	0.1757
(2019)HAN	0.1817	0.3305	0.0599	0.0496	0.0339	0.0248	0.0533	0.0874	0.1107	0.1680
(2019)DisenGCN	0.2028	0.3704	0.0665	0.0564	0.0540	0.0404	0.0943	0.1094	0.1160	0.1646
(2020)HGT	0.1743	0.3273	0.0618	0.0519	0.0401	0.0300	0.0523	0.0619	0.1135	0.1644
(2020)DGCF	<u>0.2237</u>	<u>0.4078</u>	<u>0.0998</u>	<u>0.0853</u>	0.0452	0.0329	0.1140	0.1325	0.0913	0.1400
(2021)HeCo	0.2012	0.3676	0.0598	0.0500	0.0472	0.0363	0.0950	0.1117	0.1149	0.1670
(2021)SMIN	0.1914	0.3453	0.0722	0.0584	0.0571	0.0430	0.0892	0.1076	0.1274	0.1810
(2022)RoHe	0.1926	0.3468	0.0872	0.0710	0.0543	0.0408	0.0902	0.1009	0.1300	0.1819
(2022)CGI	0.2046	0.3716	0.0586	0.0500	0.0348	0.0269	0.0762	0.0876	0.1095	0.1626
(2023)HGCL	0.2203	0.4024	0.0975	0.0807	0.0618	0.0459	0.1030	0.1210	0.1302	0.1827
(2023)HGCL+	0.2226	0.4063	0.0992	0.0821	<u>0.0628</u>	<u>0.0465</u>	<u>0.1187</u>	<u>0.1385</u>	<u>0.1323</u>	<u>0.1863</u>
(2024)RecDCL	0.1901	0.3507	0.0902	0.0737	0.0563	0.0418	0.1168	0.1355	0.1159	0.1495
(ours)BHGCL	0.2321	0.4249	0.1101	0.0941	0.0688	0.0534	0.1226	0.1446	0.1365	0.1984
Improve.	+3.76%	+4.18%	+10.32%	+10.34%	+9.55%	+14.84%	+3.28%	+4.40%	+3.17%	+6.49%
p-value.	1.37e-3	3.75e-3	1.15e-5	1.21e-5	8.92e-4	5.37e-4	6.75e-3	5.82e-3	3.56e-3	2.94e-3

The highest results are highlighted in bold, while the top baseline scores are underlined.

4.3 Experimental Setup

For the test set, unobserved items for each user are considered negative. Each algorithm predicts item rankings, excluding any items that were interacted with during the training phase. Performance is evaluated using NDCG and Recall, with K set to 10. Some of these models were initially employed for heterogeneous graph node classification tasks (e.g., HAN, HeCo, HGT). In this context, we uniformly substitute their loss functions with the BPR [30] loss, while keeping the overall framework unchanged.

For all baseline models, we prioritize using the original authors' PyTorch code or implementations through OpenHGNN [9], an open-source HGNN library. We begin by setting the hyperparameters for the baseline methods according to their original papers, and subsequently fine-tuning to achieve optimal performance. Across all models, we fix the embedding size at 128 and use a batch size of 10,240. Adam is employed as the optimizer with a learning rate of $1e^{-3}$, while parameters are initialized using the Xavier method. The values for β , λ_1 , and λ_2 are selected from $\{2e^{-4}, 5e^{-3}\}$, $\{0.0, 1e^{-5}, 1e^{-4}\}$, and $\{0.024, 0.09, 0.1\}$, respectively. Additionally, the LGC layer count is tuned within $\{1, 2, 3\}$. To prevent overfitting, we employ an early stopping criterion, which stops training when performance on the validation set, measured by recall@10, fails to improve for 20 consecutive epochs. we then select the optimal results suitable for our experimental environment.

4.4 Performance Comparison (RQ1)

The overview of the experimental outcomes, comparing model performances over multiple datasets, are summarized in Table 3. The key observations and analysis are outlined below.

- The BHGCL model outperforms all baselines on Recall@10 and NDCG@10 across all datasets, including methods based on disentangled graph-based recommenders, heterogeneous graph recommenders, contrastive learning recommenders, and IB-based recommenders.

Significance tests yielded p-values less than 0.05, confirming that the improvements are statistically significant. These gains stem from two key design factors: (1) BHGCL effectively disentangles fine-grained latent intentions from complex self-supervised signals and leverages heterogeneous semantic information from different meta-paths, aligning and enhancing their representations with intent-aware information; (2) The IB principle is applied to reduce noise in the heterogeneous graph, preserving only the minimal sufficient information for the embeddings of users and items.

- SMIN, HeCo, CGI, HGCL, and RecDCL, which incorporate contrastive learning, excel in encoding user-item interactions, particularly in sparse labeled data scenarios. By incorporating contrastive self-supervision as a complementary objective, these models enhance the process of learning representations for both users and items. In highly sparse datasets like Yelp and DoubanMovie, HGCL achieved the best baseline performance, while RecDCL, with its dual contrastive learning mechanism, excelled on DoubanBook. Graph-based disentangled models, such as DisenGCN and DGCF, alleviate data sparsity by extracting disentangled self-supervised signals. DGCF, in particular, demonstrated strong disentanglement capabilities, outperforming others on MovieLens and Amazon datasets. Additionally, attention-based models like HAN, Rohe, HERec, and NGCF+ aggregate meta-path information from neighbors. When using BPR loss, these models also show strong recommendation performance.
- Due to the potentially misleading nature of self-supervised signals, most models employing non-adaptive self-supervised contrastive learning may be misled into learning incorrect information, thereby resulting in a decline in recommendation performance. However, our BHGCL utilizes learnable graph augmentation based on the IB theory. Compared to randomly dropping elements to learn an augmented graph, our approach learns to generate more optimal and complementary enhanced user-item views, effectively capturing more comprehensive and detailed collaborative signals. Moreover, upon careful analysis of the experimental results, it becomes evident that the largest performance gain is observed on Yelp. This indicates that even in scenarios with extremely sparse data, with data density as low as 8.53×10^{-4} , BHGCL outperforms CGI in maintaining superior recommendation performance. This demonstrates that our BHGCL exhibits better stability and robustness in highly sparse data conditions.

4.5 Ablation Study (RQ2)

To understand the impact of each major components in BHGCL, ablation experiments were conducted, followed by reasonable interpretations or potential conjectures regarding the experimental results.

- *w/o - intent*: The multi-intent-aware part after LGC in the IDE (Figure 3(b)) is disabled, leaving only the LGC message propagation process.
- *w/o - disen*: The entire IDE (Figure 3(b)) is disabled, relying solely on the meta-path-based heterogeneous encoder (Figure 3(c)) with subgraphs \mathcal{G}_{uu} and \mathcal{G}_{ii} .
- *w/o - meta*: The meta-path-based heterogeneous encoder (Figure 3(c)) is excluded, using only the IDE (Figure 3(b)) for information aggregation.
- *w/o - cl*: CVCL (Figure 3(d)) is disabled, preventing alignment and enhancement of representations from Figure 3(b) and Figure 3(c).
- *w/o - ib*: In this variant, we disabled the use of the IB for regularization, i.e., Figure 3(e). Consequently, it is not possible to generate optimal learnable complementary enhanced graphs through the IB theory.

Table 4 summarizes the recommendation performance of BHGCL and its variants. The bold values represent the best performance without ablation. The comprehensive ablation results across

Table 4. Impact of Key Components on BHGCL’s Recommendation Performance

Dataset	MovieLens		Amazon		Yelp		DoubanBook		DoubanMovie	
Metrics	R@10	N@10								
w/o-intent	0.2218	0.4162	0.1058	0.0897	0.0623	0.0481	0.0931	0.1113	0.1255	0.1804
w/o-disen	0.1539	0.2941	0.0085	0.0070	0.0401	0.0300	0.0483	0.0545	0.1094	0.1608
w/o-meta	0.1871	0.3452	0.0836	0.0701	0.0530	0.0386	0.0525	0.0588	0.0936	0.1419
w/o-cl	0.2240	0.4046	0.1001	0.0849	0.0575	0.0424	0.1123	0.1312	0.0912	0.1374
w/o-ib	0.2195	0.4057	0.1009	0.0845	0.0579	0.0428	0.1216	0.1439	0.1306	0.1834
BHGCL	0.2321	0.4249	0.1101	0.0941	0.0688	0.0534	0.1226	0.1446	0.1365	0.1984

all five datasets reveal both consistent patterns and dataset-specific variations. Different datasets demonstrate varying sensitivities to component removal, which aligns with their distinct characteristics and complexity levels. First, BHGCL consistently outperforms *w/o-disen* and *w/o-meta* across all datasets, highlighting the significance of both the IDE and the SME. The impact is particularly pronounced in datasets with complex interaction patterns like Yelp and Amazon, where removing disentanglement leads to the most substantial performance drops. MovieLens, being a relatively simpler dataset, shows moderate but still significant decreases. DoubanBook and DoubanMovie demonstrate similar patterns, confirming the crucial role of these components across different recommendation scenarios. These components effectively disentangle key information from self-supervised signals and incorporate heterogeneous side information, enhancing the encoding of users and items. Notably, *w/o-intent* shows worse performance than BHGCL, indicating that relying solely on LGC within the homogeneous encoder is insufficient to capture fine-grained information from interactions. This effect varies across datasets, with more complex datasets like Yelp and Amazon showing larger performance drops compared to MovieLens, suggesting that intent modeling becomes increasingly important as the recommendation scenario grows more sophisticated. This supports our hypothesis that intent-aware factors are essential for effective CF. Additionally, BHGCL surpasses *w/o-cl*, demonstrating the benefit of cross-view knowledge transfer for improved performance. The impact of contrastive learning is notably stronger in datasets with rich heterogeneous information (Yelp, DoubanBook, and DoubanMovie) compared to simpler datasets like MovieLens. Finally, BHGCL outperforms *w/o-ib*, confirming that the IB enhances robustness by making the model more noise-resistant through IB-Regularized Optimization. The IB’s contribution varies across datasets, with its importance particularly evident in complex datasets featuring diverse user–item interactions and rich side information.

Furthermore, we conduct additional experiments to evaluate the model’s robustness under both random noise and *Meta-Attack* perturbations. As shown in Table 5, under random noise conditions, BHGCL demonstrates strong robustness with only 7.28% *average performance drop* even with 30% noise, compared to HGCL’s 15.02% decline. For meta-attacks (Table 6), BHGCL maintains remarkable stability with just 11.7% *degradation* under 30% targeted attacks, while HGCL suffers a substantial 19.2% drop. These results validate that our architecture effectively filters noise and defends against adversarial manipulations while preserving essential collaborative signals.

Through detailed ablation analysis, we observe IB mechanism enhances various downstream recommendation tasks. For *General Recommendation*, BHGCL demonstrates solid improvements over BHGCL-ib across all datasets under standard settings (0% noise). For *Sparse-Data Recommendation*, it shows particularly significant gains on sparse interaction datasets like Yelp (+18.83% Recall, +24.77% NDCG). Under *Stochastic Noise Recommendation*, Table 5 shows BHGCL maintains

Table 5. Performance Comparison Demonstrating the Effectiveness of IB Enhancement under Different Random Noise Ratios

Datasets	Noise Rate	Metrics	BHGCL-ib	BHGCL	Imp.%	
MovieLens	0%	Recall	0.2195	0.2321	5.74	-
		NDCG	0.4057	0.4249	4.73	-
	10%	Recall	0.2041	0.2212	8.38	↑
		NDCG	0.3732	0.4130	10.66	↑
	30%	Recall	0.1822	0.2049	12.46	↑
		NDCG	0.3327	0.3858	15.96	↑
Amazon	0%	Recall	0.1009	0.1101	9.12	-
		NDCG	0.0845	0.0941	11.36	-
	10%	Recall	0.0958	0.1086	13.36	↑
		NDCG	0.0798	0.0920	15.29	↑
	30%	Recall	0.0838	0.0988	17.90	↑
		NDCG	0.0695	0.0830	19.42	↑
Yelp	0%	Recall	0.0579	0.0688	18.83	-
		NDCG	0.0428	0.0534	24.77	-
	10%	Recall	0.0556	0.0680	22.30	↑
		NDCG	0.0409	0.0531	29.83	↑
	30%	Recall	0.0535	0.0672	25.61	↑
		NDCG	0.0393	0.0526	33.84	↑
DoubanBook	0%	Recall	0.1216	0.1226	0.82	-
		NDCG	0.1439	0.1446	0.49	-
	10%	Recall	0.1184	0.1215	2.62	↑
		NDCG	0.1397	0.1443	3.29	↑
	30%	Recall	0.1124	0.1152	2.49	↑
		NDCG	0.1319	0.1377	4.40	↑
DoubanMovie	0%	Recall	0.1306	0.1365	4.52	-
		NDCG	0.1834	0.1984	8.18	-
	10%	Recall	0.1288	0.1355	5.20	↑
		NDCG	0.1807	0.1964	8.69	↑
	30%	Recall	0.1222	0.1282	4.91	↑
		NDCG	0.1711	0.1895	10.75	↑

The consistent improvement percentages (Imp.%) of BHGCL over BHGCL-ib across all datasets validate the significant contribution of the IB mechanism to model robustness. “Imp.%” with bold text shows the percentage improvement of BHGCL over BHGCL-ib, and the arrow indicates the trend of improvement as noise increases.

robust performance with merely 8.2% average degradation at 30% random noise, while BHGCL-ib suffers a 15.7% decline. For *Attack-Resistant Recommendation*, Table 6 demonstrates superior defense against meta-attacks—maintaining 88.3% performance under 30% targeted perturbations compared to BHGCL-ib’s 80.8%. These comprehensive results validate that our IB mechanism is integral to enhancing various recommendation scenarios while enabling effective noise filtering and preserving essential recommendation signals.

Table 6. Performance Comparison Demonstrating Model Robustness under *Meta-Attack* Noise Ratios

Datasets	Noise Rate	Metrics	BHGCL-ib	BHGCL	Imp.%	
MovieLens	0%	Recall	0.2195	0.2321	5.74	-
		NDCG	0.4057	0.4249	4.73	-
	10%	Recall	0.1899	0.2066	8.79	↑
		NDCG	0.3193	0.3442	7.80	↑
	30%	Recall	0.1760	0.1915	8.81	↑
		NDCG	0.2633	0.2865	8.81	↑
Amazon	0%	Recall	0.1009	0.1101	9.12	-
		NDCG	0.0845	0.0941	11.36	-
	10%	Recall	0.0966	0.1081	11.90	↑
		NDCG	0.0771	0.0886	14.92	↑
	30%	Recall	0.0909	0.1026	12.87	↑
		NDCG	0.0700	0.0811	15.86	↑
Yelp	0%	Recall	0.0579	0.0688	18.83	-
		NDCG	0.0428	0.0534	24.77	-
	10%	Recall	0.0526	0.0643	22.24	↑
		NDCG	0.0385	0.0496	28.83	↑
	30%	Recall	0.0487	0.0600	23.20	↑
		NDCG	0.0337	0.0437	29.67	↑
DoubanBook	0%	Recall	0.1216	0.1226	0.82	-
		NDCG	0.1439	0.1446	0.49	-
	10%	Recall	0.1116	0.1161	4.03	↑
		NDCG	0.1250	0.1317	5.36	↑
	30%	Recall	0.0980	0.1040	6.12	↑
		NDCG	0.1006	0.1072	6.56	↑
DoubanMovie	0%	Recall	0.1306	0.1365	4.52	-
		NDCG	0.1834	0.1984	8.18	-
	10%	Recall	0.1231	0.1330	8.04	↑
		NDCG	0.1586	0.1780	12.23	↑
	30%	Recall	0.1140	0.1237	8.51	↑
		NDCG	0.1324	0.1498	13.14	↑

The consistent superior performance of BHGCL over BHGCL-ib across all datasets and noise levels validates the effectiveness of the IB mechanism in defending against meta-attacks. “Imp.%” with bold text shows the percentage improvement of BHGCL over BHGCL-ib, and the arrow indicates the trend of improvement as noise increases.

4.6 Robustness to Noisy Interactions (RQ3)

To comprehensively evaluate model robustness under random noise, we conducted detailed experiments across five datasets with varying characteristics (Amazon, DoubanBook, DoubanMovie, MovieLens, and Yelp). We introduced different levels of random perturbations (10% and 30%) into the user-item interaction patterns while maintaining the basic scale of interaction data. Figure 4 presents a comprehensive comparison of model performance under different noise levels across all

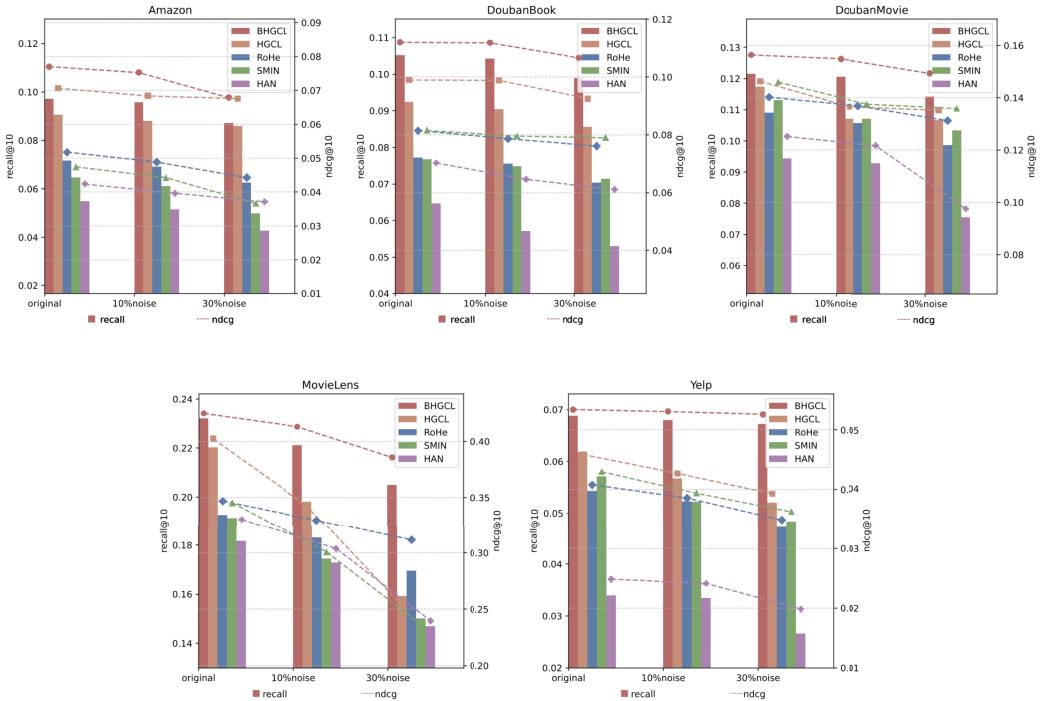


Fig. 4. Model performance comparison under different random noise levels across five datasets (Amazon, DoubanBook, DoubanMovie, MovieLens, and Yelp), with bars representing Recall@10 and solid lines depicting NDCG@10.

five datasets. The results consistently demonstrate BHGCL’s superior robustness, achieving the smallest performance degradation among all methods across diverse recommendation scenarios.

Across all five datasets, BHGCL consistently maintains the highest robustness against noise perturbations. The average performance degradation of BHGCL is remarkably small—only 1.7% drop in Recall@10 under 10% noise and 7.3% under 30% noise, substantially outperforming all baseline methods. In contrast, other methods show much larger average degradation: HGCL (13.1%), RoHe (11.3%), SMIN (15.0%), and HAN (20.2%) under 30% noise. This superior robustness is consistently observed across datasets with different characteristics. On sparse datasets like Yelp and DoubanBook, BHGCL shows exceptional stability with minimal performance drops (e.g., only 2.3% Recall@10 drop on Yelp under 30% noise). Even on dense datasets like MovieLens where noise typically has a larger impact, BHGCL maintains significantly better performance compared to baselines. This demonstrates that our BHGCL framework, leveraging an IB-based adaptive denoising technique, effectively removes useless redundant and noisy edges. By generating different denoised perspectives, BHGCL can extract the most essential *minimum sufficient information* for recommendation. The consistently superior performance of BHGCL across all datasets and noise levels validates its robust design. Whether under moderate (10%) or intensive (30%) noise conditions, BHGCL exhibits the smallest performance degradation among all methods, highlighting its exceptional ability to maintain recommendation quality despite noisy interactions. This remarkable stability fully demonstrates the potential of BHGCL in robust recommendation.

These comprehensive evaluation results clearly establish BHGCL’s outstanding robustness in handling noisy interactions. In terms of average performance degradation across all five datasets,

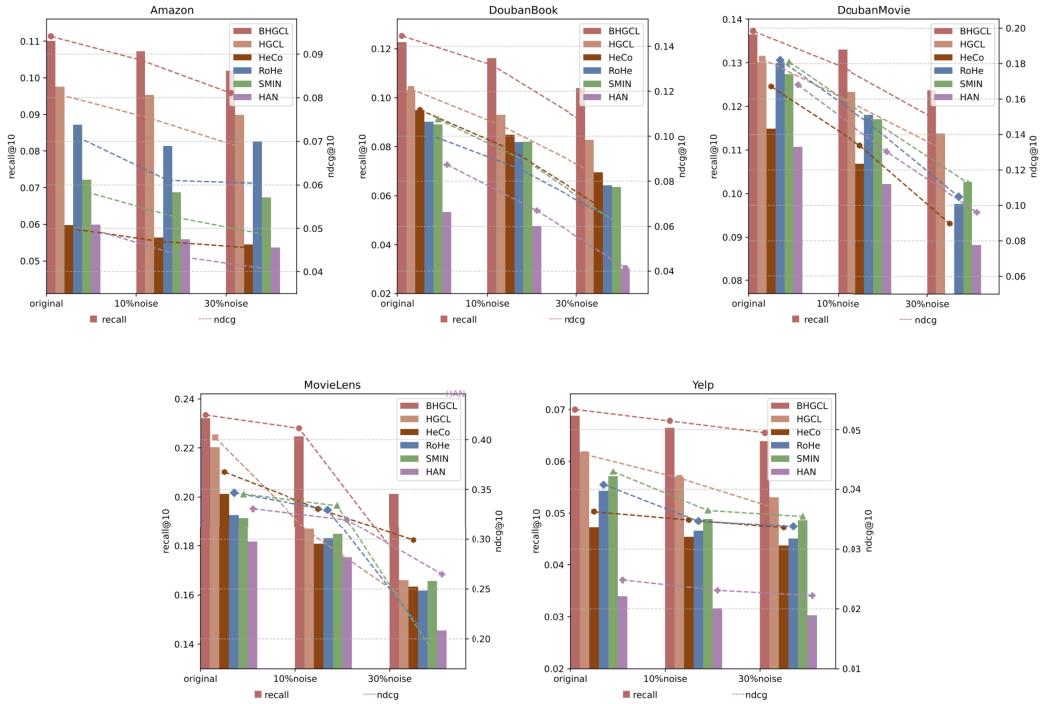


Fig. 5. Model performance comparison under different *Meta-Attack* noise levels across five datasets (Amazon, DoubanBook, DoubanMovie, MovieLens, and Yelp), with bars representing Recall@10 and solid lines depicting NDCG@10.

BHGCL exhibits substantially lower decline rates (1.7% at 10% noise and 7.3% at 30% noise) compared to the best performing baseline (5.2% and 11.3% respectively). This superior robustness, demonstrated by consistently smaller performance drops across diverse datasets and noise levels, highlights BHGCL’s significant practical advantages for real-world recommendation systems, where maintaining stable performance under varying data quality is crucial.

Furthermore, we conducted comprehensive *Meta-Attack* evaluations following recent advances in adversarial attacks on graphs [5, 15, 57, 76]. The *Meta-Attack* framework combines meta-optimization techniques with adversarial attacks, building upon pioneering work in transferable graph attacks [57, 76] and class-agnostic attack strategies [5, 15]. This enables more flexible and generalizable attack capabilities that can effectively compromise recommendation models without requiring specific model knowledge. Following [24, 39], we adapt this framework for heterogeneous recommendation scenarios through a specialized perturbation mechanism that operates on user-item interaction graphs while maintaining structural integrity.

Figure 5 presents model performance under different *Meta-Attack* intensities across five datasets. Due to space limitations, we focus our detailed analysis on DoubanBook and DoubanMovie (Tables 7 and 8), as they present particularly challenging scenarios with their rich semantic meta-paths and complex interaction structures.

On DoubanBook, BHGCL demonstrates exceptional stability under meta-attacks, with minimal degradation under 10% noise ($R@10 \downarrow 5.30\%$, $R@20 \downarrow 3.19\%$) compared to baselines like HAN ($R@10 \downarrow 10.69\%$, $N@10 \downarrow 23.34\%$). Under severe 30% attacks, while HAN experiences critical collapse ($R@10$

Table 7. Performance Comparison under Different *Meta-Attack* Noise Ratios on DoubanBook Dataset.

Dataset	DoubanBook				DoubanBook-10%				DoubanBook-30%			
	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
HAN(2019)	0.0533	0.0874	0.0771	0.0924	0.0476	0.0670	0.0699	0.0748	0.0300	0.0415	0.0447	0.0481
-	-	-	-	-	↓ 10.69%	↓ 23.34%	↓ 9.34%	↓ 19.05%	↓ 43.71%	↓ 52.52%	↓ 42.02%	↓ 47.94%
HeCo(2021)	0.0950	0.1117	0.1346	0.1196	0.0848	0.0943	0.1230	0.1038	0.0696	0.0692	0.1008	0.0785
-	-	-	-	-	↓ 10.74%	↓ 15.58%	↓ 8.62%	↓ 13.21%	↓ 26.74%	↓ 38.05%	↓ 25.11%	↓ 34.36%
SMIN(2021)	0.0892	0.1076	0.1265	0.1138	0.0821	0.0886	0.1166	0.0971	0.0636	0.0617	0.0943	0.0715
-	-	-	-	-	↓ 7.96%	↓ 17.66%	↓ 7.83%	↓ 14.68%	↓ 28.70%	↓ 42.66%	↓ 25.45%	↓ 37.17%
RoHe(2022)	0.0902	0.1009	0.1360	0.1134	0.0819	0.0857	0.1234	0.0963	0.0643	0.0643	0.0977	0.0733
-	-	-	-	-	↓ 9.20%	↓ 15.06%	↓ 9.26%	↓ 15.08%	↓ 28.71%	↓ 36.27%	↓ 28.16%	↓ 35.36%
HGCL(2023)	0.1030	0.1210	0.1499	0.1304	0.0930	0.1049	0.1385	0.1163	0.0828	0.0846	0.1256	0.0975
-	-	-	-	-	↓ 9.71%	↓ 13.31%	↓ 7.61%	↓ 10.81%	↓ 19.61%	↓ 30.08%	↓ 16.21%	↓ 25.23%
BHGCL	0.1226	0.1446	0.1755	0.1821	0.1161	0.1317	0.1699	0.1704	0.1040	0.1072	0.1546	0.1440
-	-	-	-	-	↓ 5.30%	↓ 8.90%	↓ 3.19%	↓ 6.43%	↓ 15.17%	↓ 25.83%	↓ 11.91%	↓ 20.92%

These decrease rates ↓ are calculated relative to noise ratio 0%. The highest results are highlighted in bold, while the top baseline scores are underlined.

Table 8. Performance Comparison under Different *Meta-Attack* Noise Ratios on DoubanMovie Dataset

Dataset	DoubanMovie				DoubanMovie-10%				DoubanMovie-30%			
	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
HAN(2019)	0.1107	0.1680	0.1689	0.1712	0.1022	0.1303	0.1583	0.1409	0.0882	0.0960	0.1380	0.1096
-	-	-	-	-	↓ 7.68%	↓ 22.44%	↓ 6.28%	↓ 17.70%	↓ 20.33%	↓ 42.86%	↓ 18.29%	↓ 35.98%
HeCo(2021)	0.1149	0.1670	0.1749	0.1716	0.1069	0.1337	0.1662	0.1460	0.0767	0.0897	0.1256	0.1023
-	-	-	-	-	↓ 6.96%	↓ 19.94%	↓ 4.97%	↓ 14.92%	↓ 33.25%	↓ 46.29%	↓ 28.19%	↓ 40.38%
SMIN(2021)	0.1274	0.1810	0.1913	0.1852	0.1171	0.1460	0.1785	0.1569	0.1027	0.1127	0.1605	0.1275
-	-	-	-	-	↓ 8.08%	↓ 19.34%	↓ 6.69%	↓ 15.28%	↓ 19.39%	↓ 37.73%	↓ 16.10%	↓ 31.16%
RoHe(2022)	0.1300	0.1819	0.2014	0.1976	0.1181	0.1466	0.1829	0.1593	0.0976	0.1048	0.1528	0.1186
-	-	-	-	-	↓ 9.15%	↓ 19.41%	↓ 9.19%	↓ 19.38%	↓ 24.92%	↓ 42.39%	↓ 24.13%	↓ 40.00%
HGCL(2023)	0.1302	0.1827	0.1954	0.1898	0.1233	0.1580	0.1876	0.1699	0.1138	0.1321	0.1731	0.1463
-	-	-	-	-	↓ 5.30%	↓ 13.52%	↓ 4.00%	↓ 10.48%	↓ 12.60%	↓ 27.70%	↓ 11.41%	↓ 22.92%
BHGCL	0.1365	0.1984	0.2044	0.2018	0.1330	0.1780	0.2013	0.1867	0.1237	0.1498	0.1872	0.1620
-	-	-	-	-	↓ 2.56%	↓ 10.28%	↓ 1.52%	↓ 7.48%	↓ 9.38%	↓ 24.50%	↓ 8.41%	↓ 19.72%

These decrease rates ↓ are calculated relative to noise ratio 0%. The best results are shown in bold and the second-best results are underlined.

↓ 43.71%) and SMIN shows significant deterioration (N@10 ↓ 42.66%), BHGCL maintains strong performance (R@10 = 0.1040) with consistent degradation patterns (↓ 11.91% in R@20).

For DoubanMovie's heterogeneous network with rich semantic meta-paths and diverse interaction types, BHGCL achieves remarkable stability through its synergistic combination of IB, contrastive learning and hierarchical attention. Under 10% meta-attacks, it shows minimal degradation (R@10 ↓ 2.56%, R@20 ↓ 1.52%), significantly outperforming HGCL (↓ 5.30–13.52%). Under 30% attacks, while HeCo suffers severe degradation (N@10 ↓ 46.29%), BHGCL maintains strong performance preservation (R@10 = 0.1237, N@10 = 0.1498).

These results validate BHGCL's robustness under strategic meta-attacks, demonstrating its effectiveness in preserving essential recommendation patterns through the bottleneck-enhanced architecture. The framework shows exceptional stability across diverse scenarios with different sparsity levels and meta-path structures, maintaining high recommendation quality even under severe attack conditions.

To validate IB's general applicability in enhancing robustness, we integrate it into HGCL (denoted as HGCL+) and conduct comprehensive experiments across different noise levels. As shown in Figure 6, while IB consistently improves model robustness across different architectures, its effectiveness varies significantly. When integrated into HGCL, IB brings meaningful average

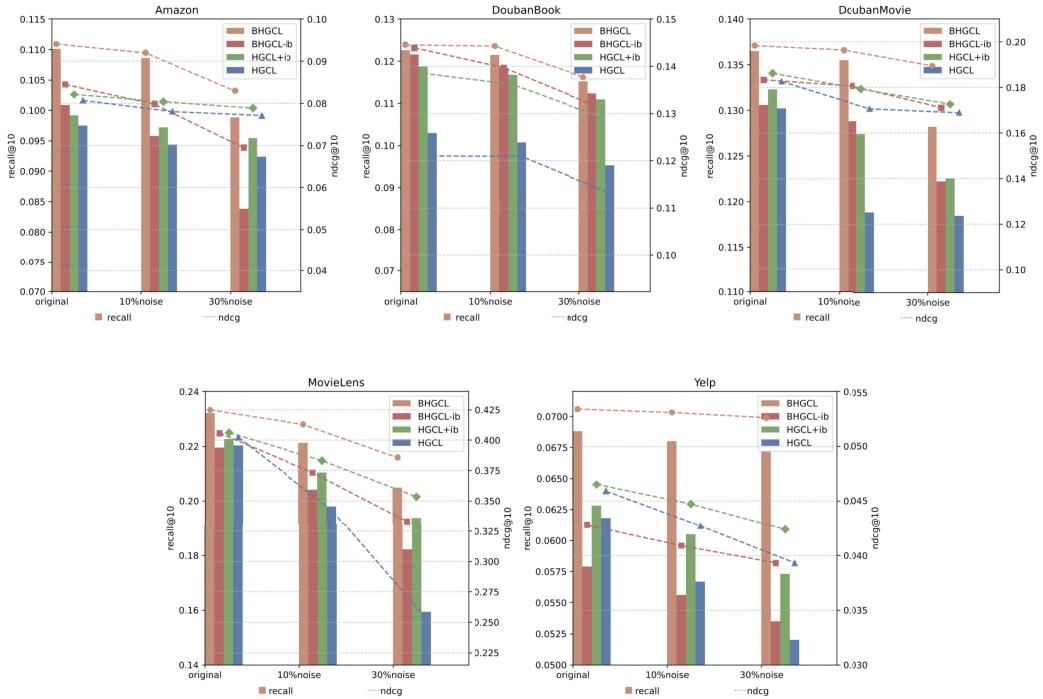


Fig. 6. Model performance comparison under different random noise levels across five datasets (Amazon, DoubanBook, DoubanMovie, MovieLens, and Yelp), with bars representing Recall@10 and solid lines depicting NDCG@10.

improvements (4.17%, 7.15%, and 11.94% under 0%, 10%, and 30% noise, respectively). However, these gains are notably smaller than when IB is incorporated into BHGCL, which achieves more substantial average improvements (8.86%, 11.89%, and 14.67%, respectively). These results demonstrate that while IB generally enhances model robustness, its effectiveness is maximized within BHGCL’s architecture. We attribute this superior synergy to BHGCL’s components that create cleaner semantic spaces for IB to operate on, enabling more effective information filtering and preservation.

4.7 Visualizing User and Item Representations

To more intuitively compare the embeddings learned by BHGCL and representative baselines, t-SNE [44] was utilized to reduce the dimensionality of the user and item embeddings, projecting them into a two-dimensional space (Figure 7). The results clearly show that BHGCL brings related users and items from the test set into close proximity after projection. This indicates that BHGCL effectively pulls the potential interaction items of users to nearby spatial positions, demonstrating the model’s capacity for accurately aligning users with relevant items. The visualization experiments here should be able to explain the experiments in Section 4.6, which demonstrate why BHGCL performs well even with added noise.

From Figure 7, it can also be observed that items of the same color in (b) HGCL, (e) SMIN, and (e) HAN are relatively clustered. This suggests that the node classification model like HAN performs well in clustering. However, there is still a gap compared to BHGCL. On the other hand, (c) RoHe, and (d) HeCo show more uniform performance but are less effective in clustering.

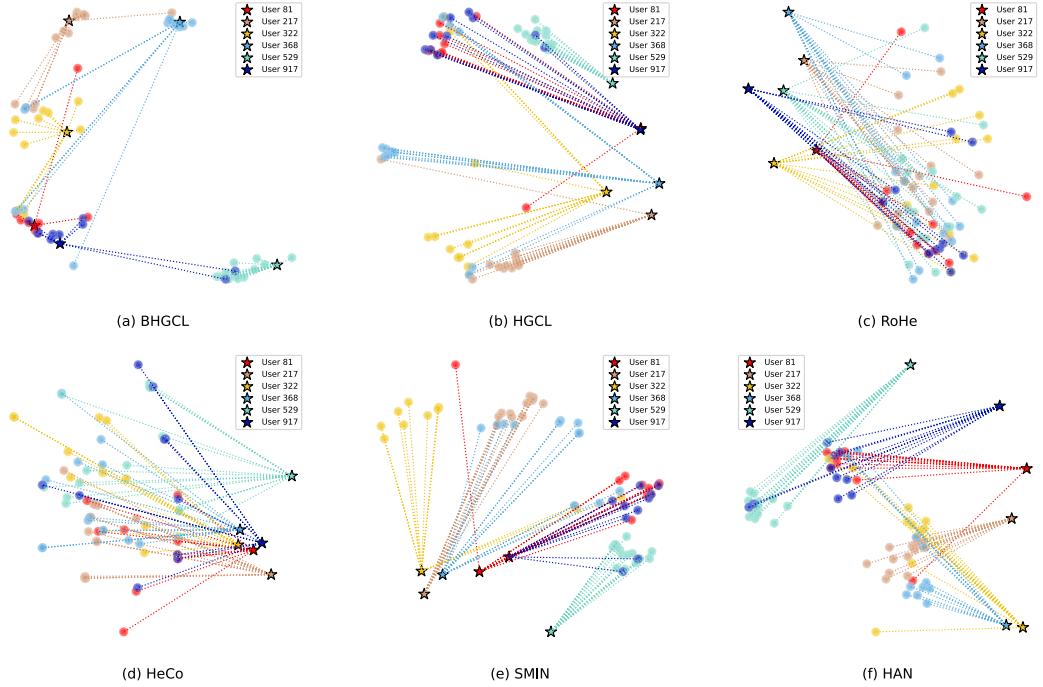


Fig. 7. Visualize the embeddings learned from BHGCL and representative baselines using t-SNE for dimensionality reduction. Users are marked by stars, and items by circles, both drawn from the test set. None of the items were linked to users during training. Points with the same color indicate related items.

4.8 Capability to Mitigate Popularity Bias (RQ4)

This section assesses how effectively various methods mitigate popularity bias. Following earlier studies [55, 58], we divided the item sets into five groups (1–5) based on popularity across three different datasets. A higher GroupID corresponds to a greater item degree within the group, signifying increased item popularity. For each dataset, we partition the overall recall@20 metric into the contributions of five distinct groups. Specifically, for the g th group of items, if the items recommended by the model also appear in both the test data and the group itself, they contribute to the g th group's recall@20.

The experimental outcomes are illustrated in Figure 8. As observed, the recommendation metrics for the fifth group are the highest across all datasets, suggesting a bias towards recommending popular items. In contrast, the likelihood of recommending less popular long-tail items is significantly lower. A closer analysis reveals that although BHGCL performs slightly worse than top-performing models in groups of two and three on Amazon, it generally captures better information about less popular items, thus mitigating the long-tail distribution problem. Particularly on the sparser DoubanBook dataset, BHGCL significantly outperforms the compared baselines across all five groups.

4.9 Kernel Density Estimation (KDE) and KDE on Angles

To further intuitively demonstrate the advantages of IB Contrastive Learning and explain the phenomena observed in Section 4.8, we employed nonparametric Gaussian KDE following [68]. Specifically, we randomly sampled 12,000 learned representations when the methods reached their

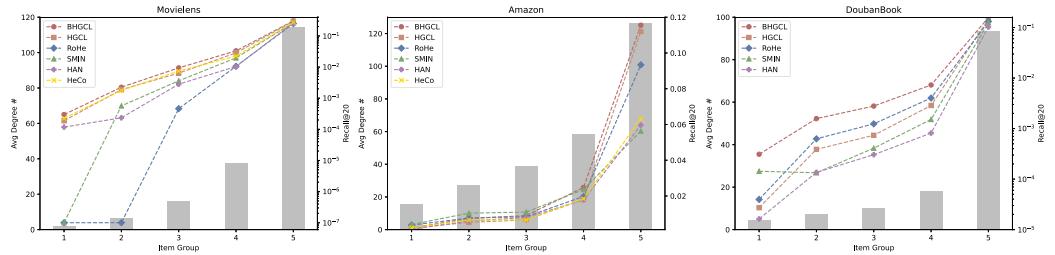


Fig. 8. Performance comparison of various item groups across the three datasets. The histograms represent the average degree of items in each group, with higher GroupIDs indicating higher degrees. The line charts represent the recall@20 metrics for the five groups for each model.

best performance and mapped them onto a unit hypersphere S^1 via t-SNE [44]. Next, we used KDE to visualize the feature distributions in \mathbb{R}^2 and analyze the angles of each point on S^1 . Figure 9 presents the results. The findings from Section 4.8 are further clarified through KDE. Highly clustered features, indicated by the denser green areas along certain arc segments, suggest that node embeddings become locally similar, potentially due to the embeddings gravitating towards popular items. This reduces feature learning for less popular items, leading to representation degeneration [27] and failing to mitigate popularity bias.

In Figure 9, compared to our (a) BHGCL, the user embeddings in (b) HGCL and (e) SMIN exhibit more pronounced clustering, as shown by sharper peaks in the angular KDE plots. This suggests that these models are more dominated by popular items, which could reduce generalization ability, as seen in Section 4.8. In contrast, BHGCL exhibits moderate clustering, leading to better generalization.

Although the embeddings of (c) RoHe, (d) HeCo, and (f) HAN show a more uniform distribution, their performance in niche item recommendation, as shown in Section 4.8, is significantly worse than BHGCL. We hypothesize that BHGCL's moderate clustering helps capture user preferences for niche items by identifying sparse but meaningful correlations. As noted in SimGCL [68], while uniformity and generalization tend to increase together, this correlation is limited to a certain range. Overabundant uniformity can overlook important interactions between similar users/items, which negatively impacts recommendation performance. The lower performance of RoHe, HeCo, and HAN, despite their uniform embeddings, suggests that this excessive uniformity fails to capture niche items, explaining why BHGCL outperforms these methods in Section 4.8.

4.10 Case Study: Intent-Semantic Pattern Analysis

Through visualization analysis as shown in Figure 10, we observe an interesting pattern where users u_{1063} and u_{2067} , despite having no direct connections, demonstrate similar intent-aware preferences. These users share strong interactions with common items (398, 1,245, 435) in category C_1 , validating our model's ability to capture meaningful semantic relationships through UIU meta-paths. Specifically, by leveraging both intent modeling and meta-path encoding, our model successfully identifies the preference similarity between these two users: their shared items in C_1 exhibit consistently high interaction weights, indicating strong intent alignment. Meanwhile, the clear category-based item grouping (C_0 , C_1 , C_2) through IGI meta-paths reveals how our framework effectively captures global semantic structures beyond local interaction patterns. This case study demonstrates how our approach combines fine-grained intent analysis with broader semantic relationships to discover meaningful user preference patterns that might be overlooked by traditional CF methods.

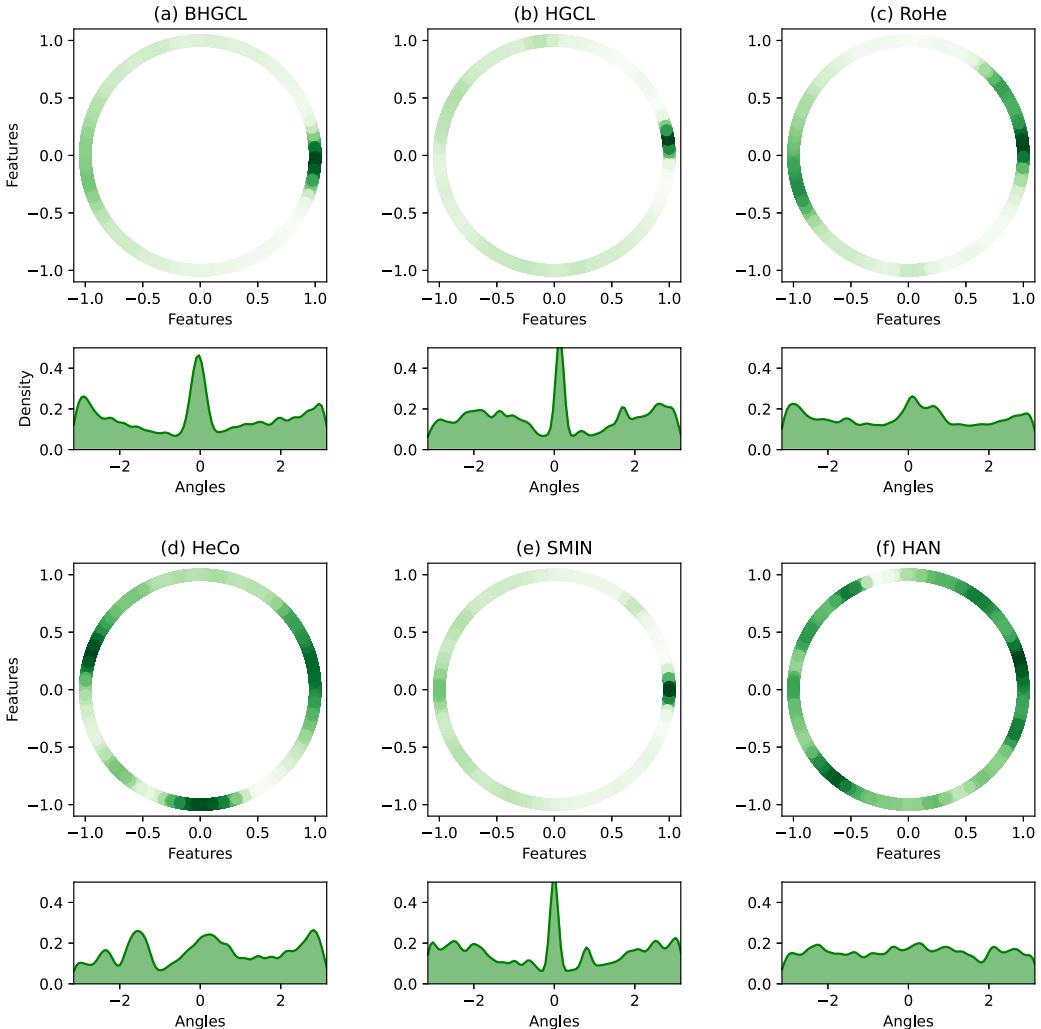


Fig. 9. KDE and KDE in angle on 12,000 randomly sampled embeddings from DoubanBook.

4.11 Hyperparameter Analysis

We analyze how variations in key hyperparameters affect performance, focusing on IB-regularized strength (β), the CVCL coefficient (λ_2), embedding dimension (d), and learning rate (lr). Figures 11 and 12 illustrate the outcomes of the experiments.

— *IB-Regularized Strength*: It is evident that as the IB-regularization strength increases (i.e., as β increases), the recommendation performance on five datasets initially rises to a peak and then declines. This is likely because when $\beta = 0$, meaning there is no IB regularization, the model may overfit the training data that contains noise. These noisy details do not exist in the test data, resulting in slightly weaker model performance. As β begins to increase and reaches an optimal value, the BHGCL effectively compresses the input information, removing redundant information and noise. This allows the model to learn more effective information relevant to the recommendation task, thereby significantly improving recommendation performance.

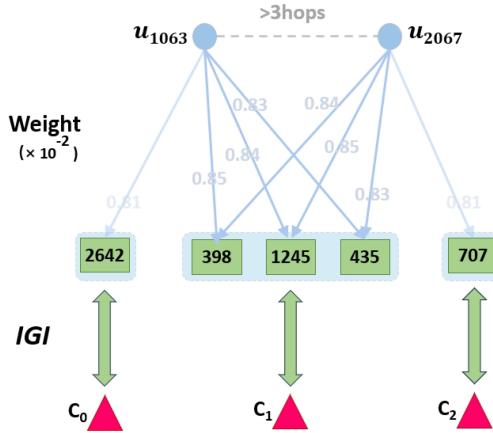


Fig. 10. Case study of intent-aware global user relations and category-based item grouping. Non-locally connected users (u_{1063} and u_{2067}) exhibit similar intent-aware preferences through overlapping item categories, demonstrated by their high-weight interactions with shared items (398, 1,245, 435). Items are grouped by categories (C_0 , C_1 , C_2) through IGI meta-paths, revealing semantic relationships beyond local collaborative signals.

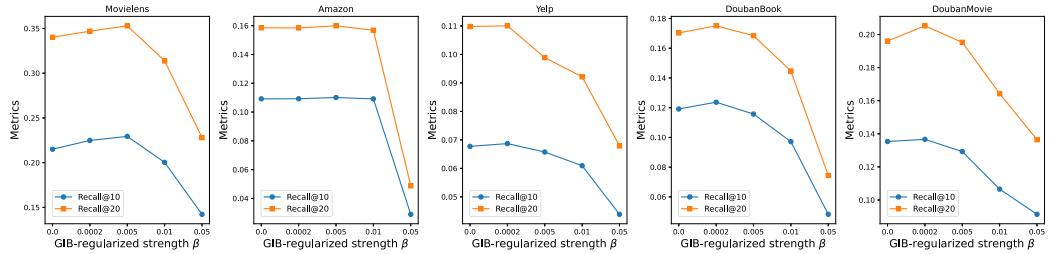


Fig. 11. Performance trends with different IB-regularized strength (β).

However, when β becomes too large, the compression becomes overly aggressive, causing the model to lose important information related to the target task. Consequently, BHGCL cannot effectively recommend in HIN, as the representations learned by BHGCL become overly simplified, failing to capture sufficient complexity, which leads to a decline in generalization ability. Upon further observation, we find that the sparser the dataset, the smaller the optimal β tends to be. One possible reason is that in sparse datasets, the original input information inherently contains less redundancy and noise, thereby reducing the need for information compression.

– *Cross-View Contrastive Learning Coefficient*: Alignment between intent-aware representations and heterogeneous side information from meta-paths is controlled by the coefficient λ_2 . The impact of λ_2 varies across datasets, but an appropriately chosen value can significantly enhance recommendation performance. By leveraging contrastive learning across different views, BHGCL can better handle data inconsistencies and noise, resulting in improved robustness and accuracy. However, if λ_2 is too small, the contrastive learning signal weakens, leading to insufficient optimization of user and item representations across views. This limited capture of multi-view semantics reduces the richness and accuracy of the representations, ultimately hindering the BHGCL’s capacity to identify key information and causing a decline in recommendation performance.

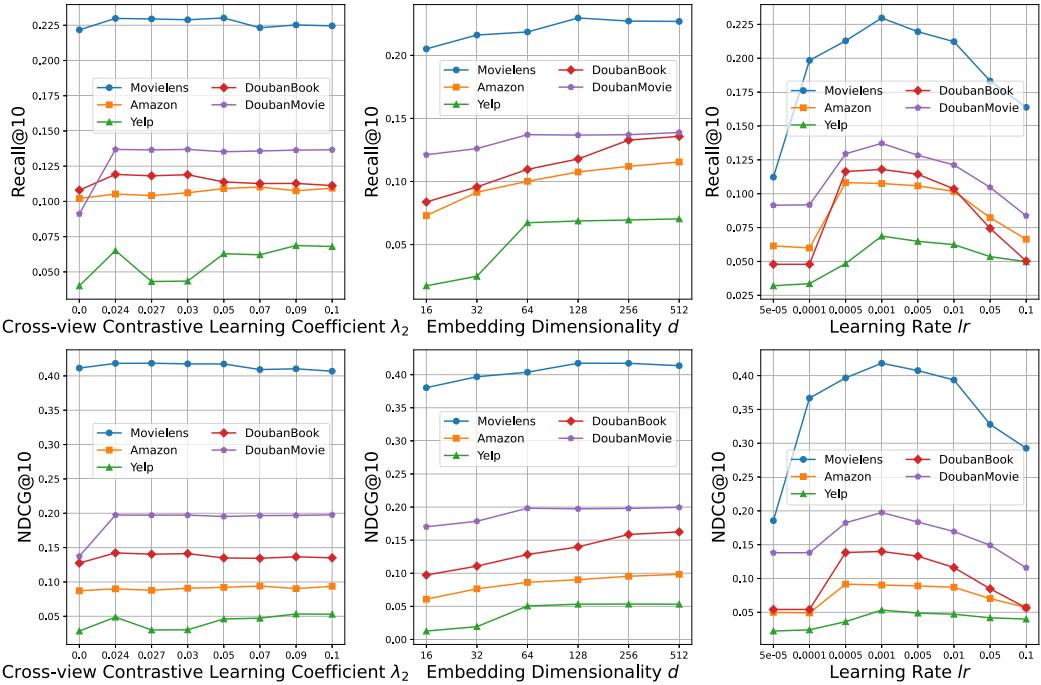


Fig. 12. Hyperparameter study of the BHGCL.

- *Embedding Dimension*: We varied the embedding dimensionality d from 16 to 512 across five datasets. It was observed that increasing the embedding dimensionality d initially improves performance, but further increases cause the performance to plateau and even decline on certain datasets. Therefore, there exists a performance saturation point for d ; beyond this point, further increases in d lead to performance degradation due to overfitting.
- *Learning Rate*: The overall performance trend across different datasets is consistent. Each dataset achieves optimal performance within a moderate learning rate range (0.0005 to 0.001), while performance deteriorates with higher learning rates. Although on certain datasets, such as Amazon, $lr = 0.0005$ exhibits the best performance, $lr = 0.001$ also yields satisfactory results. Therefore, for the sake of fairness and efficiency in comparison, we report the final experimental results using $lr = 0.001$.

5 Related Work

A comprehensive overview of recent advancements in recommender systems is provided here, emphasizing GNN-based approaches, disentangled representations, contrastive learning, and the IB principle.

5.1 GNN-Based Recommender Systems

By learning from the relationships between nodes, recent graph-based models have revolutionized recommender systems, allowing for more efficient latent representation learning. Early models like NGCF [49] pioneered multi-layer propagation for capturing high-order connectivity, while LightGCN [10] simplified GNN architectures to focus on neighbor aggregation. To address the challenge of sparse data, KGs have been integrated into GNN frameworks [49], facilitating better

modeling of complex user and item dynamics. Techniques like GraphRec and PinSage [49] exploit the power of social networks and random walks to advance large-scale recommendation systems. In HINs, researchers have proposed both meta-path-based solutions [37] and approaches that bypass the use of meta-paths to capture diverse entity interactions. These advancements demonstrate GNNs' potential in capturing complex user-item relationships, showcasing their broad applicability in recommender systems. For instance, [6] highlights *targeted shilling attacks*, where attackers manipulate recommendations for specific users by designing fake user profiles and interactions. Such attacks exploit the GNN's structure, bypassing traditional defenses. Similarly, [25] discusses *poisoning attacks* where generative models create fake users to promote specific items while maintaining overall system accuracy. To enhance diversity, [62] proposes *DGRec*, which employs submodular neighbor selection and attention mechanisms to balance diversity and accuracy in GNN-based recommendations. The challenge of scalability is addressed by *Linear-Time Graph Neural Networks* in [73], which optimizes GNN computation to match the efficiency of matrix factorization methods. Additionally, collaboration-aware models like *CAGCN* [54] improve prediction accuracy by focusing on more meaningful user-item interactions, while also enhancing computational efficiency.

5.2 Recommendation via Disentangled Representations

The quest for interpretability and efficiency in recommender systems has sparked interest in disentangled representation learning. This approach aims to isolate distinct latent factors within data, potentially enhancing both model performance and explainability. Initial strides in this direction were made with methods like Graph Attention Networks, which implicitly disentangle node representations through attention mechanisms. DisenGCN [22] advanced this concept by explicitly separating latent factors in graph structures via a neighborhood routing mechanism. In domains like Point-of-Interest systems, distinguishing between sequential dynamics and spatial influences has proven essential for improving performance. For instance, DisenPOI [26] proposes a dual-graph framework that disentangles sequential and geographical influences using contrastive learning, yielding significant improvements in recommendation accuracy. In the recommender systems domain, DGCF [52] introduced multiple latent factors to elucidate user-item interactions, while models like DisenHAN have applied disentangled attention to heterogeneous graphs, yielding notable improvements in recommendation accuracy and interpretability. Recent research has further refined disentangled representations by incorporating curriculum learning and leveraging KGs. For example, DR-MTCDR [7] explores disentangled representation learning to handle multiple recommendation targets across different domains to transfer trustworthy domain-shared information efficiently. Similarly, DIKGNN [42] proposes a novel approach that utilizes financial domain knowledge to deliver personalized fund recommendations, effectively addressing the unique challenges of the financial industry. However, challenges persist, particularly in HGNNs and scenarios with sparse user-item interactions. To address these issues, ongoing research is exploring cross-domain self-supervised learning techniques, which may offer new pathways to more robust and informative disentangled representations in recommender systems. CDR [47] leverages a causal graph to address the sparse and shifting nature of user preferences, resulting in recommendations that generalize more effectively across varying contexts.

5.3 Contrastive Learning for Recommendation

Self-supervised contrastive learning has become a prominent method for enhancing the robustness and quality of representations used in recommender systems. Several notable works have demonstrated its effectiveness through various data augmentation and feature extraction strategies. For instance, SGL [58] employs multi-view data augmentation via random node and edge dropping,

while NCL [20] enriches contrastive learning by incorporating neighborhood information in graph CF. HCCF [60] utilizes a hypergraph-based approach to model collaborative relationships at both the local and global levels. MHCH [67] advances social recommendation systems by incorporating a multi-channel hypergraph, and STGCN [72] models complex user-item dynamics through reconstructed graph convolutions. Sequential, multi-behavior, and multi-interest recommendation scenarios have benefited from the application of contrastive learning methods. However, most existing approaches are limited to homogeneous user-item graphs. While some studies have introduced self-supervision into heterogeneous graph representation learning [29, 53], they primarily focus on node classification rather than recommendation scenarios. More recent advances, such as RecDCL [71], propose a dual contrastive learning framework that combines BCL and FCL. This framework addresses redundant solutions in user-item representation learning, leading to improved recommendation accuracy. Similarly, GAECL [61] focuses on graph augmentation for contrastive learning in CF, addressing noisy data by reconstructing the adjacency matrix using structural optimization and PageRank centrality to generate augmented views that filter out unimportant topologies and attributes. By extending contrastive learning to multi-relational graphs, the RCL model [56] effectively models user interactions that reflect both immediate changes and more sustained behavioral trends across multiple types of behaviors. Meanwhile, the AdaGCL framework [17] employs two adaptive contrastive view generators—one for generating graphs and the other for denoising—to dynamically tailor data augmentation, addressing the challenges posed by noisy and imbalanced data distributions in real-world recommendation scenarios. Another contribution, KACL [46], integrates KG data into recommendation systems, addressing challenges such as interaction domination and knowledge overload. It creates distinct views for interaction data and KG information, and applies contrastive learning to capture shared patterns while filtering out irrelevant edges. Finally, RMCL [64] proposes a review-based multi-intention contrastive learning approach that models user review intentions based on a mixed Gaussian distribution hypothesis and establishes fine-grained connections between user and item reviews through a multi-intention contrastive strategy, significantly improving performance in real-world recommendation scenarios.

5.4 Learning via IB Principle

Rooted in information theory, the IB framework [41] serves as an effective method for producing concise and meaningful representations by discarding noise that is irrelevant to the task at hand, achieved through the use of information-theoretic regularization. The IB principle has demonstrated its effectiveness across various domains, including graph representation learning. For instance, it has been successfully applied to homogeneous graphs, as shown in [59], where the Graph Information Bottleneck method effectively regularizes both graph structure and node features. This concept has subsequently been extended to heterogeneous graphs [63]. Inspired by the IB principle, recent works [55, 75] have applied IB theory to recommendation systems, particularly on bipartite graphs. Recent studies have further investigated the application of the IB concept in heterogeneous and evolving graph structures. Yuan et al. [70] introduced the DGIB model, which focuses on generating effective representations for such graphs. Their method incorporates the MSC criterion to reduce unnecessary data while retaining critical features within latent spaces. Yang et al. [65] proposed the GBSR model, aiming to address noisy social networks in recommendation systems. This approach applies the IB framework to refine social structures by optimizing the information shared between the processed graph and recommendation outcomes, while reducing its correlation with the original graph. The PMIB algorithm, introduced by Hu et al. [13], is a clustering algorithm that requires no parameters. This technique automatically detects important details from multiple sources, optimizing multi-view clustering tasks without the need for manual involvement. This approach demonstrates the versatility of the IB principle in handling multi-modal data without

the need for parameter tuning. The IB approach consistently emphasizes balancing the tradeoff between maximizing the retention of relevant information (reconstruction term) and minimizing the complexity of the representation (minimality term), thereby achieving accurate predictions while filtering out redundant noise across various graph-based learning scenarios.

6 Conclusion

This study presents a novel framework called BHGCL tailored for HINs recommendation tasks. BHGCL strengthens robust recommendation by initially performing feature comparison of disentangled representations and embeddings extracted from different meta-paths. This approach absorbs heterogeneous auxiliary data, refining the embeddings and ensuring their alignment. A graph augmentation method, inspired by the IB framework, is then employed to selectively eliminate data noise and highlight informative structures, thereby capturing more effective and informative collaborative signals. Evaluation across multiple datasets with varying characteristics demonstrates that BHGCL provides superior accuracy and exhibits stronger robustness compared to current leading approaches.

References

- [1] Desheng Cai, Shengsheng Qian, Quan Fang, and Changsheng Xu. 2022. Heterogeneous hierarchical feature aggregation network for personalized Micro-Video recommendation. *IEEE Transactions on Multimedia* 24 (2022), 805–818. DOI: <https://doi.org/10.1109/TMM.2021.3059508>
- [2] Chong Chen, Weizhi Ma, Min Zhang, Zhaowei Wang, Xiuqiang He, Chenyang Wang, Yiqun Liu, and Shaoping Ma. 2021. Graph heterogeneous multi-relational recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, 5 (May), 3958–3966. DOI: <https://doi.org/10.1609/aaai.v35i5.16515>
- [3] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, 1 (April 2020), 27–34. DOI: <https://doi.org/10.1609/aaai.v34i1.5330>
- [4] Mengru Chen, Chao Huang, Lianghao Xia, Wei Wei, Yong Xu, and Ronghua Luo. 2023. Heterogeneous graph contrastive learning for recommendation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining (WSDM '23)*. ACM, New York, NY, 544–552. DOI: <https://doi.org/10.1145/3539597.3570484>
- [5] Weiwei Feng, Baoyuan Wu, Tianzhu Zhang, Yong Zhang, and Yongdong Zhang. 2021. Meta-attack: Class-agnostic and model-agnostic physical adversarial attack. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 7767–7776. DOI: <https://doi.org/10.1109/ICCV48922.2021.00769>
- [6] Sihan Guo, Ting Bai, and Weihong Deng. 2023. Targeted shilling attacks on GNN-based recommender systems. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*. ACM, New York, NY, 649–658. DOI: <https://doi.org/10.1145/3583780.3615073>
- [7] Xiaobo Guo, Shaoshuai Li, Naicheng Guo, Jiangxia Cao, Xiaolei Liu, Qiongxu Ma, Runsheng Gan, and Yunan Zhao. 2023. Disentangled representations learning for multi-target cross-domain recommendation. *ACM Trans. Inf. Syst.* 41, 4, Article 85 (March 2023), 27. Pages. DOI: <https://doi.org/10.1145/3572835>
- [8] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. Yee Whye Teh and Mike Titterington (Eds.), PMLR, 297–304. Retrieved from <https://proceedings.mlr.press/v9/gutmann10a.html>
- [9] Hui Han, Tianyu Zhao, Cheng Yang, Hongyi Zhang, Yaoqi Liu, Xiao Wang, and Chuan Shi. 2022. OpenHGNN: An open source toolkit for heterogeneous graph neural network. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM '22)*. ACM, New York, NY, 3993–3997. DOI: <https://doi.org/10.1145/3511808.3557664>
- [10] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. ACM, New York, NY, 639–648. DOI: <https://doi.org/10.1145/3397271.3401063>
- [11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. DOI: <https://doi.org/10.1145/3038912.3052569>

- [12] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 2018. Leveraging meta-path based context for Top- N recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. ACM, New York, NY, 1531–1540. DOI: <https://doi.org/10.1145/3219819.3219965>
- [13] Shizhe Hu, Rulin Geng, Zhaoxu Cheng, Chaoyang Zhang, Guoliang Zou, Zhengzheng Lou, and Yangdong Ye. 2022. A parameter-free multi-view information bottleneck clustering method by cross-view weighting. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22)*. ACM, New York, NY, 3792–3800. DOI: <https://doi.org/10.1145/3503161.3547985>
- [14] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020 (WWW '20)*. ACM, New York, NY, 2704–2710. DOI: <https://doi.org/10.1145/3366423.3380027>
- [15] Jincheng Huang, Lun Du, Xu Chen, Qiang Fu, Shi Han, and Dongmei Zhang. 2023. Robust mid-pass filtering graph convolutional networks. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*. ACM, New York, NY, 328–338. DOI: <https://doi.org/10.1145/3543507.3583335>
- [16] Eric Jiang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *Proceedings of the International Conference on Learning Representations*. Retrieved from <https://openreview.net/forum?id=rkE3y85ee>
- [17] Yangqin Jiang, Chao Huang, and Lianghao Huang. 2023. Adaptive graph contrastive learning for recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*. ACM, New York, NY, 4252–4261. DOI: <https://doi.org/10.1145/3580305.3599768>
- [18] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR '17)*. Retrieved from <https://openreview.net/forum?id=SJU4ayYgl>
- [19] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37. DOI: <https://doi.org/10.1109/MC.2009.263>
- [20] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM Web Conference 2022 (WWW '22)*. ACM, New York, NY, 2320–2329. DOI: <https://doi.org/10.1145/3485447.3512104>
- [21] Xiaoling Long, Chao Huang, Yong Xu, Huance Xu, Peng Dai, Lianghao Xia, and Liefeng Bo. 2021. Social recommendation with Self-Supervised metagraph informax network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM '21)*. ACM, New York, NY, 1160–1169. DOI: <https://doi.org/10.1145/3459637.3482480>
- [22] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning*. Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.), PMLR, 4212–4221. Retrieved from <https://proceedings.mlr.press/v97/ma19a.html>
- [23] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. *Learning Disentangled Representations for Recommendation*. Curran Associates Inc., Red Hook, NY, USA.
- [24] Yao Ma, Suhang Wang, Tyler Derr, Lingfei Wu, and Jiliang Tang. 2021. Graph adversarial attack via rewiring. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)*. ACM, New York, NY, 1161–1169. DOI: <https://doi.org/10.1145/3447548.3467416>
- [25] Toan Nguyen Thanh, Nguyen Duc Khang Quach, Thanh Tam Nguyen, Thanh Trung Huynh, Viet Hung Vu, Phi Le Nguyen, Jun Jo, and Quoc Viet Hung Nguyen. 2023. Poisoning GNN-based recommender systems with generative surrogate-based attacks. *ACM Trans. Inf. Syst.* 41, 3, Article 58 (February 2023), 24 pages. DOI: <https://doi.org/10.1145/3567420>
- [26] Yifang Qin, Yifan Wang, Fang Sun, Wei Ju, Xuyang Hou, Zhe Wang, Jia Cheng, Jun Lei, and Ming Zhang. 2023. DisenPOI: Disentangling sequential and geographical influence for point-of-interest recommendation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining (WSDM '23)*. ACM, New York, NY, 508–516. DOI: <https://doi.org/10.1145/3539597.3570408>
- [27] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining (WSDM '22)*. ACM, New York, NY, 813–823. DOI: <https://doi.org/10.1145/3488560.3498433>
- [28] Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. 2023. Disentangled contrastive collaborative filtering. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. ACM, New York, NY, 1137–1146. DOI: <https://doi.org/10.1145/3539618.3591665>
- [29] Yuxiang Ren, Bo Liu, Chao Huang, Peng Dai, Liefeng Bo, and Jiawei Zhang. 2019. Heterogeneous deep graph infomax. arXiv:1911.08538. Retrieved from <http://arxiv.org/abs/1911.08538>
- [30] Steffen Rendle, Christoph Freudenthaler, Zeno Gantert, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, 452–461.

- [31] Lei Sang, Weichen Fei, Yi Zhang, Yuee Huang, and Yiwen Zhang. 2024. Heterogeneous adaptive preference learning for recommendation. *ACM Transactions on Recommender Systems* (2024). DOI: <https://doi.org/10.1145/3656480>
- [32] Lei Sang, Honghao Li, Yiwen Zhang, Yi Zhang, and Yun Yang. 2024. AdaGIN: Adaptive graph interaction network for click-through rate prediction. *ACM Trans. Inf. Syst.* 43, 1 (2024), 1–31. DOI: <https://doi.org/10.1145/3681785>
- [33] Lei Sang, Yu Wang, Yiwen Zhang, and Xindong Wu. 2025. Denoising heterogeneous graph pre-training framework for recommendation. *ACM Trans. Inf. Syst.* 43, 5, Article 121 (July 2025), 31 pages. DOI: <https://doi.org/10.1145/3706632>
- [34] Lei Sang, Yu Wang, Yi Zhang, Yiwen Zhang, and Xindong Wu. 2025. Intent-guided heterogeneous graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 37, 4 (2025), 1915–1929. DOI: <https://doi.org/10.1109/TKDE.2025.3536096>
- [35] Lei Sang, Min Xu, Shengsheng Qian, Matt Martin, Peter Li, and Xindong Wu. 2021. Context-dependent propagating-based video recommendation in multimodal heterogeneous information networks. *IEEE Trans. Multimed.* 23 (2021), 2019–2032. DOI: <https://doi.org/10.1109/TMM.2020.3007330>
- [36] Lei Sang, Min Xu, Shengsheng Qian, and Xindong Wu. 2023. Adversarial heterogeneous graph neural network for robust recommendation. *IEEE Trans. Comput. Soc. Syst.* 10, 5 (2023), 2660–2671. DOI: <https://doi.org/10.1109/TCSS.2023.3268683>
- [37] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S. Yu Philip. 2018. Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowl. Data Eng.* 31, 2 (2018), 357–370.
- [38] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. 2019. Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowl. Data Eng.* 31, 2 (February 2019), 357–370. DOI: <https://doi.org/10.1109/TKDE.2018.2833443>
- [39] Lichao Sun, Yingtong Dou, Carl Yang, Kai Zhang, Ji Wang, Philip S. Yu, Lifang He, and Bo Li. 2023. Adversarial attack and defense on graph data: A survey. *IEEE Trans. Knowl. Data Eng.* 35, 08 (August 2023), 7693–7711. DOI: <https://doi.org/10.1109/TKDE.2022.3201243>
- [40] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. PathSim: Meta path-based top-K similarity search in heterogeneous information networks. *Proc. VLDB Endow.* 4, 11 (August 2011), 992–1003. DOI: <https://doi.org/10.14778/3402707.3402736>
- [41] Naftali Tishby, Fernando C. Pereira, and William Bialek. 2000. The information bottleneck method. arXiv:physics/0004057. Retrieved from <https://arxiv.org/abs/physics/0004057>
- [42] Ke Tu, Wei Qu, Zhengwei Wu, Zhiqiang Zhang, Zhongyi Liu, Yiming Zhao, Le Wu, Jun Zhou, and Guannan Zhang. 2023. Disentangled interest importance aware knowledge graph neural network for fund recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*. ACM, New York, NY, 2482–2491. DOI: <https://doi.org/10.1145/3583780.3614846>
- [43] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. arXiv:1807.03748. Retrieved from <http://arxiv.org/abs/1807.03748>
- [44] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, 86 (2008), 2579–2605. Retrieved from <http://jmlr.org/papers/v9/vandermaaten08a.html>
- [45] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards representation alignment and uniformity in collaborative filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. ACM, New York, NY, 1816–1825. DOI: <https://doi.org/10.1145/3534678.3539253>
- [46] Hao Wang, Yao Xu, Cheng Yang, Chuan Shi, Xin Li, Ning Guo, and Zhiyuan Liu. 2023. Knowledge-adaptive contrastive learning for recommendation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining (WSDM '23)*. ACM, New York, NY, 535–543. DOI: <https://doi.org/10.1145/3539597.3570483>
- [47] Wenjie Wang, Xinyu Lin, Liuhui Wang, Fuli Feng, Yunshan Ma, and Tat-Seng Chua. 2023. Causal disentangled recommendation against user preference shifts. *ACM Trans. Inf. Syst.* 42, 1, Article 12 (August 2023), 27 pages. DOI: <https://doi.org/10.1145/3593022>
- [48] Xin Wang, Hong Chen, Yuwei Zhou, Jianxin Ma, and Wenwu Zhu. 2023. Disentangled representation learning for recommendation. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 1 (2023), 408–424. DOI: <https://doi.org/10.1109/TPAMI.2022.3153112>
- [49] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19)*. ACM, New York, NY, 165–174. DOI: <https://doi.org/10.1145/3331184.3331267>
- [50] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Chua Tat-Seng. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the Web Conference 2021 (WWW '21)*. ACM, New York, NY, 878–887. DOI: <https://doi.org/10.1145/3442381.3450133>
- [51] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous graph attention network. In *Proceedings of the World Wide Web Conference (WWW '19)*. ACM, New York, NY, 2022–2032. DOI: <https://doi.org/10.1145/3308558.3313562>

- [52] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. ACM, New York, NY, 1001–1010. DOI: <https://doi.org/10.1145/3397271.3401137>
- [53] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. 2021. Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)*. ACM, New York, NY, 1726–1736. DOI: <https://doi.org/10.1145/3447548.3467415>
- [54] Yu Wang, Yuying Zhao, Yi Zhang, and Tyler Derr. 2023. Collaboration-aware graph convolutional network for recommender systems. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*. ACM, New York, NY, 91–101. DOI: <https://doi.org/10.1145/3543507.3583229>
- [55] Chunyu Wei, Jian Liang, Di Liu, and Fei Wang. 2022. Contrastive graph structure learning via information bottleneck for recommendation. In *Advances in Neural Information Processing Systems*. S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 20407–20420. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2022/file/803b9c4a8e4784072fd791c54d614e2-Paper-Conference.pdf
- [56] Wei Wei, Lianghao Xia, and Chao Huang. 2023. Multi-relational contrastive learning for recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*. ACM, New York, NY, 338–349. DOI: <https://doi.org/10.1145/3604915.3608807>
- [57] Wojciech Widel, Preetam Mukherjee, and Mathias Ekstedt. 2022. Security countermeasures selection using the meta attack language and probabilistic attack graphs. *IEEE Access* 10 (2022), 89645–89662. DOI: <https://doi.org/10.1109/ACCESS.2022.3200601>
- [58] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. ACM, New York, NY, 726–735. DOI: <https://doi.org/10.1145/3404835.3462862>
- [59] Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. 2020. Graph information bottleneck. In *Advances in Neural Information Processing Systems*. H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 20437–20448. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2020/file/ebc2aa04e75e3caabda543a1317160c0-Paper.pdf
- [60] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. ACM, New York, NY, 70–79. DOI: <https://doi.org/10.1145/3477495.3532058>
- [61] Lixiang Xu, Yusheng Liu, Tong Xu, Enhong Chen, and Yuanyan Tang. 2024. Graph augmentation empowered contrastive learning for recommendation. *ACM Trans. Inf. Syst.* 43, 2 (2024), 1–27. DOI: <https://doi.org/10.1145/3677377>
- [62] Liangwei Yang, Shengjie Wang, Yunzhe Tao, Jiankai Sun, Xiaolong Liu, Philip S. Yu, and Taiqing Wang. 2023. DGRec: Graph neural network for recommendation with diversified embedding generation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining (WSDM '23)*. ACM, New York, NY, 661–669. DOI: <https://doi.org/10.1145/3539597.3570472>
- [63] Liang Yang, Fan Wu, Zichen Zheng, Bingxin Niu, Junhua Gu, Chuan Wang, Xiaochun Cao, and Yuanfang Guo. 2021. Heterogeneous graph information bottleneck. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI '21)*. Zhi-Hua Zhou (Ed.), International Joint Conferences on Artificial Intelligence Organization, 1638–1645. DOI: <https://doi.org/10.24963/ijcai.2021/226>
- [64] Wei Yang, Tengfei Huo, Zhiqiang Liu, and Chi Lu. 2023. Review-based multi-intention contrastive learning for recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. ACM, New York, NY, 2339–2343. DOI: <https://doi.org/10.1145/3539618.3592053>
- [65] Yonghui Yang, Le Wu, Zihan Wang, Zhuangzhuang He, Richang Hong, and Meng Wang. 2024. Graph bottlenecked social recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*. ACM, New York, NY, 3853–3862. DOI: <https://doi.org/10.1145/3637528.3671807>
- [66] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, Quoc Nguyen, and Hung Viet. 2021. Socially-aware self-supervised tri-training for recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)*. ACM, New York, NY, 2084–2092. DOI: <https://doi.org/10.1145/3447548.3467340>
- [67] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Quoc Nguyen, Hung Viet, and Zhang Xiangliang. 2021. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *Proceedings of the Web Conference 2021 (WWW '21)*. ACM, New York, NY, 413–424. DOI: <https://doi.org/10.1145/3442381.3449844>
- [68] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 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 (SIGIR '22)*. ACM, New York, NY, 1294–1303. DOI: <https://doi.org/10.1145/3477495.3531937>

- [69] Wei Yu and Shijun Li. 2018. Recommender systems based on multiple social networks correlation. *Future Generation Computer Systems* 87, C (October 2018), 312–327. DOI: <https://doi.org/10.1016/j.future.2018.04.079>
- [70] Haonan Yuan, Qingyun Sun, Xingcheng Fu, Cheng Ji, and Jianxin Li. 2024. Dynamic graph information bottleneck. In *Proceedings of the ACM Web Conference 2024 (WWW '24)*. ACM, New York, NY, 469–480. DOI: <https://doi.org/10.1145/3589334.3645411>
- [71] Dan Zhang, Yangliao Geng, Wenwen Gong, Zhongang Qi, Zhiyu Chen, Xing Tang, Ying Shan, Yuxiao Dong, and Jie Tang. 2024. RecDCL: Dual contrastive learning for recommendation. In *Proceedings of the ACM Web Conference 2024 (WWW '24)*. ACM, New York, NY, 3655–3666. DOI: <https://doi.org/10.1145/3589334.3645533>
- [72] Jian Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. 2019. STAR-GCN: Stacked and reconstructed graph convolutional networks for recommender systems. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI '19)*. AAAI Press, 4264–4270.
- [73] Jiahao Zhang, Rui Xue, Wenqi Fan, Xin Xu, Qing Li, Jian Pei, and Xiaorui Liu. 2024. Linear-time graph neural networks for scalable recommendations. In *Proceedings of the ACM Web Conference 2024 (WWW '24)*. ACM, New York, NY, 3533–3544. DOI: <https://doi.org/10.1145/3589334.3645486>
- [74] Mengmei Zhang, Xiao Wang, Meiqi Zhu, Chuan Shi, Zhiqiang Zhang, and Zhou Jun. 2022. Robust heterogeneous graph neural networks against adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 4 (June 2022), 4363–4370. DOI: <https://doi.org/10.1609/aaai.v36i4.20357>
- [75] Q. Zhang, L. Xia, X. Cai, S. Yiu, C. Huang, and C. S. Jensen. 2024. Graph augmentation for recommendation. In *Proceedings of the 2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE Computer Society, 557–569. DOI: <https://doi.org/10.1109/ICDE60146.2024.00049>
- [76] Daniel Zügner, Oliver Borchert, Amir Akbarnejad, and Stephan Günnemann. 2020. Adversarial attacks on graph neural networks: Perturbations and their patterns. *ACM Trans. Knowl. Disc. Data* 14, 5, Article 57 (June 2020), 31 pages. DOI: <https://doi.org/10.1145/3394520>

Received 28 September 2024; revised 19 May 2025; accepted 19 July 2025