This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3536096



# Intent-guided Heterogeneous Graph Contrastive Learning for Recommendation

Lei Sang, Yu Wang, Yi Zhang, Yiwen Zhang*, Xindong Wu *Fellow, IEEE*

***Abstract*—Contrastive Learning (CL)-based recommender systems have gained prominence in the context of Heterogeneous Graph (HG) due to their capacity to enhance the consistency of representations across different views. However, existing frameworks often neglect the fact that user-item interactions within HG are governed by diverse latent intents (e.g., brand preferences or demographic characteristics of item audiences), which are pivotal in capturing fine-grained relations. The exploration of these underlying intents, particularly through the lens of meta-paths in HGs, presents us with two principal challenges: i) How to integrate CL with intents; ii) How to mitigate noise from meta-path-driven intents.**

**To address these challenges, we propose an innovative framework termed *Intent-guided Heterogeneous Graph Contrastive Learning* (IHGCL), which designed to enhance CL-based recommendation by capturing the intents contained within meta-paths. Specifically, the IHGCL framework includes: i) a meta-path-based Dual Contrastive Learning (DCL) approach to effectively integrate intents into the recommendation, constructing intent-intent contrast and intent-interaction contrast; ii) a Bottlenecked AutoEncoder (BAE) that combines mask propagation with the information bottleneck principle to significantly reduce noise perturbations introduced by meta-paths. Empirical evaluations conducted across six distinct datasets demonstrate the superior performance of our IHGCL framework relative to conventional baseline methods. Our model implementation is available at https://github.com/wangyu0627/IHGCL.**

***Index Terms*—Recommendation, Heterogeneous Graph Neural Networks, Contrastive Learning, Intent Modeling, Information Bottleneck**

## I. Introduction

Recommender systems [1], [2] play an increasingly crucial role in daily life, including content delivery in short videos [3], news [4], and shopping [5]. These systems use users' implicit historical interactions to effectively assist them in discovering items or products that align with their preferences [6]. Traditional recommendation methods [5], [7] often overlook the potential value of users' auxiliary attributes and items' label features. These methods rely solely on collaborative filtering [7], [8] through the user-item interaction graph [5] to infer user preferences. However, in many real-world scenarios, the available interaction data are typically highly sparse, posing challenges for these traditional approaches.

Lei Sang, Yu Wang, Yi Zhang and Yiwen Zhang, are with School of Computer Science and Technology, Anhui University 230601, Hefei, Anhui, China. E-mail: sanglei@ahu.edu.cn, wangyuahu@stu.ahu.edu.cn, zhangyi@stu.ahu.edu.cn, zhangyiwen@ahu.edu.cn.

Xindong Wu is with the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, Hefei 230601, Anhui, P.R. China. E-mail: xwu@hfut.edu.cn

*Corresponding author.

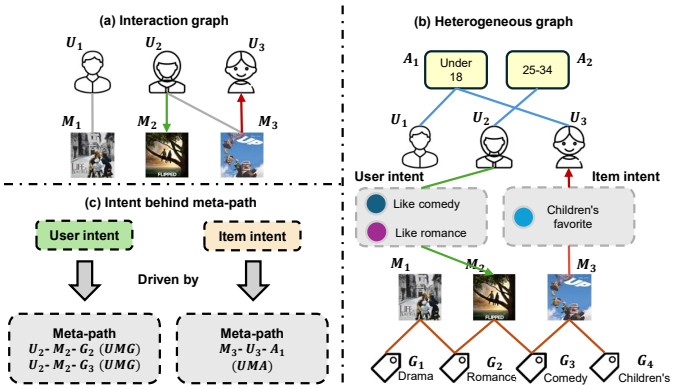


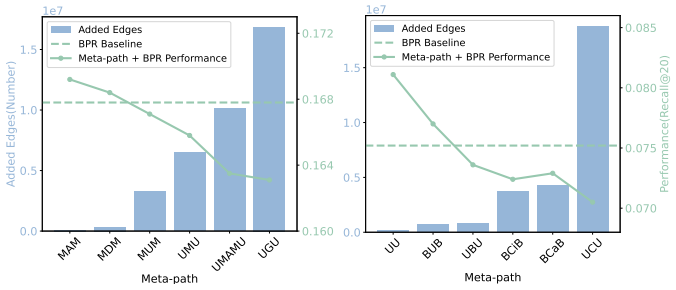

Fig. 1: (a) Interaction graph in a movie scenario, where the green arrow indicates recommending movie $M_2$ to user $U_2$, and the red arrow indicates user $U_3$ to movie $M_3$; (b) heterogeneous graph incorporating user and item intents, showing that interactions are guided by intents; (c) considering the intents driven by the meta-paths.

Contrastive Learning (CL)-based recommendation models [9]–[12] have emerged as a novel perspective to address the limitations of data sparsity. In contrast to traditional collaborative filtering techniques, these methods aim to maximize the consistency of representations across different views to capture more valuable supervision signals. CL-based recommendation models employ the principles of alignment and uniformity [13] to spread node embeddings apart and maximize the information entropy in the embedding space. This ensures that similar nodes are positioned closely together, while dissimilar nodes are distant from each other. In essence, the embeddings of users and items should be distributed in the space in a tight and dispersed manner. For instance, SimGCL [11] and SGL [10] utilize Gaussian noise and graph augmentation, respectively, to implement this concept. Building upon these advancements, CL-based models have been further adapted to the more complex heterogeneous graph (HG) recommendation scenarios [3], [14], where multiple types of nodes and relationships are modeled. For example, HGCL [15] constructs contrastive views by combining HG-based auxiliary information with user-item interactions. These recommendation models have demonstrated promising performance in various settings.

Despite their effectiveness and explainability, to the best of our knowledge, these studies largely overlook the underlying fine-grained intents of users and items. In real-life applications, the formation of user-item interactions is driven by many intent factors [16], [17], such as purchasing skincare products based on specific skin concerns like dryness or sensitivity. Taking

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3536096






(a) Douban Movie [23] (b) Yelp [24]

Fig. 2: The impact w.r.t. different meta-paths. The blue bar is the number of added edges current meta-path, and the green line indicates the performance for current meta-path.

Figure 1 as an example, the part (a) represents the bipartite graph recommendation paradigm, while the part (b) illustrates the modeling user and item intents from existing information. Meta-paths [18], [19] are commonly used in heterogeneous graph as tools for capturing preferences between nodes. For example, the intent between user $U_2$ and movie $M_2$ is driven by the meta-paths '$U_2 - M_2 - G_2$' and '$U_2 - M_2 - G_3$'. We then consider recommending $M_1$ (belongs to $G_2$), or $M_3$ (belongs to $G_3$), to user $U_2$. Similarly, the intent between movie $M_3$ and user $U_3$ is the meta-path '$M_3 - U_3 - A_1$'. In this case, recommending movie watched by user $U_1$ (age group $A_1$) is more suitable. These observations underscore the profound influence of respective intents on both users and items. Heterogeneous Graph Neural Network (HGNN) [19], [20] can effectively capture richer and deeper behaviors and preferences of users and items from complex data structures by meta-path based aggregation. This motivates us to employ this approach to capture the intents behind meta-paths to enhance CL-based recommendation. Therefore, we have the following two challenges to address:

**C1: How to integrate CL with intents?** Existing heterogeneous graph recommendation methods [15], [20], [21] have investigated the effectiveness of contrastive learning in transferring information. These methods design contrastive objectives between the auxiliary heterogeneous information and the main user-item view. However, they have not constructed multiple contrastive views based on the intents behind meta-paths. Multi-view contrasts have been proven to be vital in improving recommendation performance [9], [10], [22]. The rich semantic information from meta-paths contains more fine-grained user preferences and item features, which are powerful tools for capturing intents. Overall, modeling users' and items' intents and integrating them into the main user-item view to construct contrastive objectives is a meaningful challenge.

**C2: How to mitigate noise from meta-path-driven intents?** In heterogeneous graph, meta-path-based intents are used to construct new connections between users or items, which help alleviate data sparsity and improve recommendation accuracy. However, it is challenging to distinguish whether these connections are informative or merely noise. As illustrated in Figure 2, we construct user-user or item-item interaction graphs using a single meta-path at each time to improve recommendation under the BPR [6] loss. Experimental results demonstrate that when a specific type of meta-path introduces more connections, it may also lead to a decline in recommendation performance. Therefore, we hypothesize that although meta-paths bring rich semantic information, they also contain indescribable noise [25], [26]. This noise propagated through the layer of HGNN to farther nodes diminishes the recommendation performance.

To tackle the challenges mentioned above, we propose an Intent-guided Heterogeneous Graph Contrastive Learning (**IHGCL**) framework for recommendation. First, for **C1**, a meta-path enhanced **Dual Contrastive Learning** (DCL) is proposed, which aims to align the different intent-based embeddings learned by users and items. Intent-intent contrast unifies user and item preferences by maximizing the consistency of intent embeddings. Intent-interaction contrast involves adding intent embeddings to the representation of real interactions to perform representation-level data augmentation. For **C2**, we design a **Bottlenecked Autoencoder** (BAE) to mitigate the noise issues induced by meta-paths. BAE reconstructs robust and denoised node representations via a dual-masked autoencoder, with the masking operation applied to the node embeddings. It also introduces an information bottleneck loss to constrain the amount of information between the autoencoder and the graph structure, aiming to find the **minimum sufficient** representation of a dataset.

Compared to BIGCF [27], our method enhances intent modeling with meta-paths, addressing its coarse-grained and sparse representations. Meanwhile, unlike LightGCL [9], which focuses on global relationships, we capture fine-grained user-item intents through meta-path-guided contrastive learning. Furthermore, we resolve the semantic misalignment issues in HGCL [21] by leveraging hidden intents in heterogeneous data for more precise and effective recommendations. Overall, IHGCL is an effective framework for HG-based recommendation that considers the potential of intents behind meta-paths for constructing contrastive views and effectively mitigates the noise issues introduced by intents. The contributions are summarized as follows:

- We explore the intents behind meta-paths and integrate them into contrastive learning to construct views, and propose the recommendation framework IHGCL, which models the intents of users and items via a Dual Contrastive Learning (DCL) module.
- We further propose a Bottlenecked Autoencoder (BAE), which effectively mitigates the noise issues introduced by the meta-paths, while adaptively preserving the topological structure through information bottleneck techniques.
- We conduct extensive experiments on six public datasets, and the results show that IHGCL not only outperforms existing baselines but also demonstrates effectiveness from various aspects.

## II. Preliminaries

### A. Definitions

In this section, we formally define some significant concepts related to heterogeneous graph as follows:

**Heterogeneous Graph (HG [19]):** A HG is characterized by a graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}, \phi, \varphi)$, where the

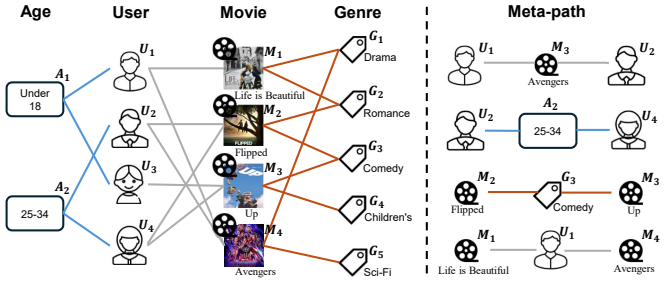



Fig. 3: A toy example of heterogeneous graph on Movielens [28] dataset for recommendation.

collections of nodes and edges are symbolized by $\mathcal{V}$ and $\mathcal{E}$, respectively. For each node $v$ and edge $e$, associated type mapping functions exist: $\phi : \mathcal{V} \to \mathcal{A}$ for nodes and $\varphi : \mathcal{E} \to \mathcal{R}$ for edges. Here, $\mathcal{A}$ represents the node types and $\mathcal{R}$ denotes the edge types, with the total number of types exceeding two, i.e., $|\mathcal{A}| + |\mathcal{R}| > 2$.

**Meta-path [18]:** In a $\mathcal{G}$, a meta-path $\rho$ is represented as $\mathcal{A}_1 \xrightarrow{\mathcal{R}_1} \mathcal{A}_2 \xrightarrow{\mathcal{R}_2} \ldots \xrightarrow{\mathcal{R}_l} \mathcal{A}_{l+1}$, describing a composite connection between $\mathcal{A}_1$ and $\mathcal{A}_{l+1}$. Figure 3 shows that we establish connections between users (or items) from the heterogeneous graph on the left according to the given meta-paths. For instance, users $U_1$ and $U_2$ can be connected by the path '$U_1 - M_3 - U_2$' to enrich the data. Typically, different meta-paths reveal varied dependency information between two nodes. The path '$UAU$' indicates that two users belong to the same age group, while '$MGM$' suggests that two movies belong to the same genre.

**Meta-path-based Subgraph:** We define a specific meta-path $\rho \in \mathcal{R}$ and the corresponding node type $V \in \mathcal{A}$. For a node $v \in V$, the set of edges formed by connecting $v^\rho$ to new nodes through the meta-path $\rho$ are denoted as $e$. By traversing other nodes in $V$, we obtain a set of edges $E$. Consequently, $E$ and the set of all nodes of this specific type (denoted by $V$) constitute a subgraph $G_V^\rho$. Taking Figure 3 as an instance, we define $\rho$ as "$MGM$" and the node type $V$ as "movie". All movie-movie connections formed through this meta-path constitute the meta-path-based subgraph $G_M^{MGM}$.

### B. Problem Formulation

Typical recommendation scenarios include a set of $M$ users $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ and a set of $N$ items $\mathcal{I} = \{i_1, i_2, \ldots, i_N\}$. Furthermore, historical user-item interaction records are stored in a matrix $\mathbf{R}^{M \times N}$. Some research papers [5], [7] define a bipartite graph structure:

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^\top & \mathbf{0} \end{pmatrix}, \tag{1}$$

Let the initialized embedding be $\mathbf{E}^{(0)} \in \mathbb{R}^{(M+N) \times d}$, where $d$ is the embedding dimension. $\mathbf{E}^{(0)}$ includes $\mathbf{E}_u^{(0)}$ and $\mathbf{E}_i^{(0)}$, and through normalization propagation [5], we obtain the $l$-th layer embeddings:

$$\mathbf{E}^{(l+1)} = \left(\mathbf{D}^{-0.5}\mathbf{A}\mathbf{D}^{-0.5}\right)\mathbf{E}^{(l)}, \tag{2}$$

where $\mathbf{D}$ is a degree matrix used to measure the number of non-zero entries in each row of $\mathbf{A}$. The existing methods for contrastive learning can generally be categorized into data-based (DA), feature-based (FA), and model-based (MA) augmentations [29]:

$$\begin{cases} \text{DA:} & \mathbf{A}^{'} = \mathcal{T}^{'}(\mathbf{A}), \quad \mathbf{A}^{''} = \mathcal{T}^{''}(\mathbf{A}) \\ \text{FA:} & \mathbf{E}^{'} = \mathbf{E} + \Delta^{'}, \quad \mathbf{E}^{''} = \mathbf{E} + \Delta^{''} \\ \text{MA:} & \mathbf{Z}^{'} = \mathcal{F}^{'}(\mathbf{A}, \mathbf{E}), \quad \mathbf{Z}^{''} = \mathcal{F}^{''}(\mathbf{A}, \mathbf{E}) \end{cases} \tag{3}$$

where $\mathcal{T}$ is the graph structure transformer, such as random edge dropout and structural learning. $\Delta$ typically represents noise with a specified distribution. $\mathcal{F}$ is defined as a view generator containing learnable parameters. For example, an intent disentanglement module used to capture user intent. The common practice in CL-based recommendation [9]–[11] involves generating two augmented views combined with the main view for contrast, which goes against the existing paradigm [15], [21] of heterogeneous contrastive recommendation (aligning side information with the main task). We delve into this paradigm in our motivation and propose a solution.

## III. METHODOLOGY

The overview of the proposed IHGCL is shown in Figure 4. We propose a model-based augmentation for contrastive learning, which leverages the intents in heterogeneous information to model the user and item intents. The model comprises into four modules: model input, a bottlenecked autoencoder, dual contrastive learning, and model optimization.

### A. Model Input

The input to the model is divided into the main user-item view and the auxiliary heterogeneous information from the user view and item view.

- **User-item view.** We follow the classic CL-based recommendation task [10], [11], using $\mathbf{A}$ mentioned in Section II-B and embeddings $\mathbf{E}_u$ and $\mathbf{E}_i$ for users and items as the main view's input. $\mathbf{E}_u$ and $\mathbf{E}_i$ serve as shared parameters in the parallel training process.
- **Heterogeneous view.** To construct multi-view augmentations for contrastive learning, we model the preferences of user $u$ and attributes of item $i$, which can be achieved through meta-paths [15], [23]. A specific meta-path that connects two nodes embodies a similar semantic relation, that is, the user's group and the item's category. Based on the Section II, we respectively select two meta-path-based subgraphs of users and items for the model:

$$\textbf{User} : \boldsymbol{G}_U^{\rho_1^u}, \boldsymbol{G}_U^{\rho_2^u}; \quad \textbf{item} : \boldsymbol{G}_I^{\rho_1^i}, \boldsymbol{G}_I^{\rho_2^i}, \tag{4}$$

where $\rho_k^u$ denotes the $k$-th meta-path based on users, and the items are the same definition.

The five graphs on the left in Figure 4, along with the embeddings of users and items, constitute the input of the model. Such an operation often brings a substantial amount of interaction data to alleviate sparsity in recommendation. However, we have demonstrated in Figure 2 that such subgraphs usually contain noise [25], [26].

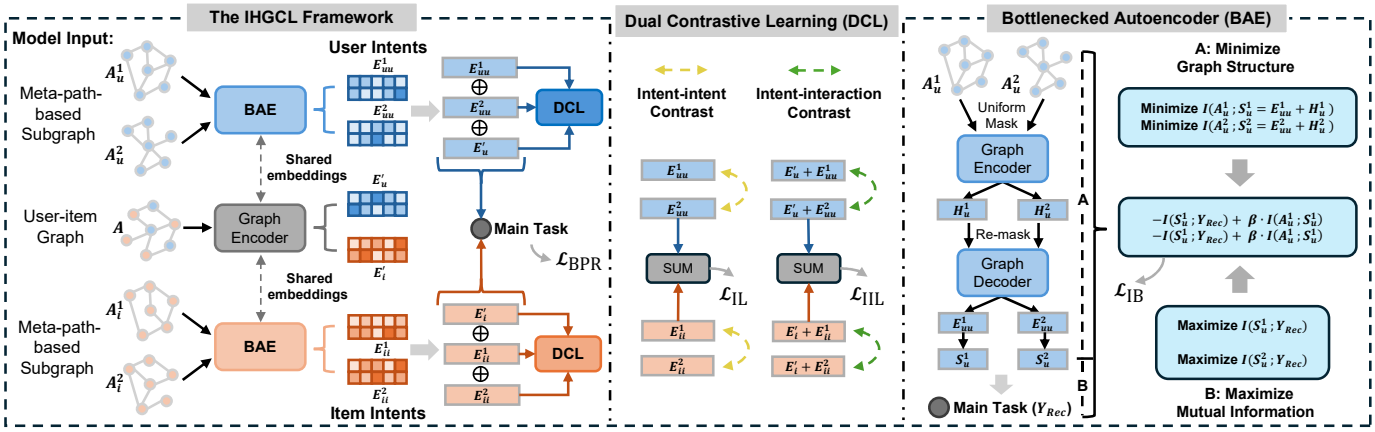



Fig. 4: The complete framework of the proposed IHGCL, which consists of a Dual Contrastive Learning (DCL) module and a Bottlenecked Autoencoder (BAE). The DCL module generates two types of contrasts: contrasts between meta-paths and contrasts between meta-path-enhanced views, providing self-supervised signals. The BAE module employs a dual-masked autoencoder combined with an adaptive information bottleneck technique to mitigate the noise issues, which can capture the **minimum sufficient information** from the data features.

### *B. Bottlenecked Autoencoder (BAE)*

In this section, we introduce the BAE to model rich intents and mitigate the noise in heterogeneous information. The masked autoencoder can better reconstruct noise-polluted data by learning the global structure of the data. The information bottleneck method improves reconstruction accuracy by retaining **minimum sufficient** information to the recommendation.

Given the rich intents from meta-paths, we need to preserve the user preferences and item attributes to the greatest extent without randomly disrupting the data structure. Many studies [20], [30] have demonstrated that the similar intents of each meta-path can be treated as a distinct view, and for each view, we employ the same BAE. First, the model defines meta-path-based subgraphs of users and items as:

$$\boldsymbol{G}_U^j = (\boldsymbol{V}_u, \mathbf{A}_u^j, \mathbf{E}_u), \quad \boldsymbol{G}_I^j = (\boldsymbol{V}_i, \mathbf{A}_i^j, \mathbf{E}_i) \quad \text{for } j = 1, 2 \tag{5}$$

where $\mathbf{A}_u^j$ and $\mathbf{A}_i^j$ are the matrices corresponding to Eq. 4, and $j$ represents the $j$-th meta-path of user or item. $|\boldsymbol{V}_u| = M$ and $|\boldsymbol{V}_i| = N$ are the number of users and items. Next, we employ a sampling strategy without replacement to obtain the node set $\boldsymbol{V}^{[Mask]}$ with masks before entering the encoder:

$$\begin{aligned} \boldsymbol{V}_u^{[Mask]} &= \{\boldsymbol{V}_s \in \boldsymbol{V}_u \mid r_s \le p\}, \quad r_s \sim \text{Uniform}(0,1), \\ \boldsymbol{V}_i^{[Mask]} &= \{\boldsymbol{V}_s \in \boldsymbol{V}_i \mid r_s \le p\}, \quad r_s \sim \text{Uniform}(0,1), \end{aligned} \tag{6}$$

where $p$ is the mask ratio, and $\boldsymbol{V}_s$ is the node set based on $r_s$. To mitigate the risk of sampling biases, where a node's entire neighbor is either fully masked or fully visible, uniform random sampling is employed. This technique enhances the encoder's generalization ability by avoiding localized bias centers. We mask the entire node embeddings in the set $\boldsymbol{V}^{[Mask]}$ to obtain $\widetilde{\mathbf{E}}_u = \{\mathbf{e}_u^1, \mathbf{e}_u^2, \ldots, \mathbf{e}_u^M\}$ and $\widetilde{\mathbf{E}}_i = \{\mathbf{e}_i^1, \mathbf{e}_i^2, \ldots, \mathbf{e}_i^N\}$. Next, we empirically employ LightGCN [5] as the encoder to perform convolution on the masked embeddings:

$$\begin{aligned} \mathbf{h}_u^{(L_E)} &= \sum_{v \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_v|}} \mathbf{h}_v^{(L_E-1)}, \\ \mathbf{h}_i^{(L_E)} &= \sum_{j \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i||\mathcal{N}_j|}} \mathbf{h}_j^{(L_E-1)}, \end{aligned} \tag{7}$$

where $L_E$ is the encoder layer, and $\mathbf{h}_v^{(0)}$ and $\mathbf{h}_j^{(0)}$ denote one of the nodes $\widetilde{\mathbf{E}}_u$ and $\widetilde{\mathbf{E}}_i$. $\mathcal{N}_u$ and $\mathcal{N}_i$ represent the first-order receptive fields of users and items, respectively. The first reconstructed embeddings are $\mathbf{H}_u = \{\mathbf{h}_u^1, \mathbf{h}_u^2, \ldots, \mathbf{h}_u^M\}$ and $\mathbf{H}_i = \{\mathbf{h}_i^1, \mathbf{h}_i^2, \ldots, \mathbf{h}_i^N\}$. From a macro perspective, this is a matrix multiplication operation: $\left(\mathbf{D}_u^{-0.5}\mathbf{A}_u\mathbf{D}_u^{-0.5}\right)^{L_E}\widetilde{\mathbf{E}}_u$ and $\left(\mathbf{D}_i^{-0.5}\mathbf{A}_i\mathbf{D}_i^{-0.5}\right)^{L_E}\widetilde{\mathbf{E}}_i$, where $\mathbf{D}_u$ and $\mathbf{D}_i$ are the diagonal degree matrices of $\mathbf{A}_u$ and $\mathbf{A}_i$ ($\mathbf{A}_u \in \mathbb{R}^{M\times M}$ and $\mathbf{A}_i \in \mathbb{R}^{N\times N}$).

Considering that nodes aggregate information from a subset of their neighbors, these operations reduce dependency on specific nodes. However, when the masking rate is low, the node embeddings may still contain direct information from the original input features (i.e., noise), leading to the failure of model-based augmentations. Therefore, we adopt the re-masking and decoder to address this issue. BAE employs the same sampling strategy to obtain a new set $\boldsymbol{V}^{[Remask]}$ for masking the reconstructed embeddings to obtain $\widetilde{\mathbf{H}}_u$ and $\widetilde{\mathbf{H}}_i$. The construction of the encoder and decoder is identical, except that the input embeddings are re-masked:

$$\begin{aligned} \mathbf{E}_{uu} &= \left(\mathbf{D}_u^{-0.5}\mathbf{A}_u\mathbf{D}_u^{-0.5}\right)^{L_D}\widetilde{\mathbf{H}}_u, \\ \mathbf{E}_{ii} &= \left(\mathbf{D}_i^{-0.5}\mathbf{A}_i\mathbf{D}_i^{-0.5}\right)^{L_D}\widetilde{\mathbf{H}}_i, \end{aligned} \tag{8}$$

where $L_D$ is the decoder layer. $\mathbf{E}_{uu}$ and $\mathbf{E}_{ii}$ are the final outputs of BAE. Expanding to multiple heterogeneous views and their inputs, we can obtain $\mathbf{E}_{uu}^1$, $\mathbf{E}_{uu}^2$, $\mathbf{E}_{ii}^1$ and $\mathbf{E}_{ii}^2$.

We use GNNs [7], [31] as encoders and decoders due to their strong capability to aggregate neighborhood information and iteratively propagate it. Moreover, LightGCN typically

performs this task well, thereby enhancing the efficiency of recommendation. However, such autoencoders overly rely on the masked nodes, which leads to insufficient generalization capability. Thus, we apply the Information Bottleneck (IB) [32], [33] to constrain the encoder's and decoder's embeddings, which limits the amount of information between the autoencoder and the graph structure, thereby forcing the model to focus on the essential features of the data. This method enhances the model's robustness to noise. First, we adopt the loss function $\mathcal{L}_{\text{IB}}$ between the autoencoder's embeddings and graph structure. Specifically, we define $(\mathbf{E}_{uu} + \mathbf{H}_u)$ and $(\mathbf{E}_{ii} + \mathbf{H}_i)$ as $\mathbf{S}_u$ and $\mathbf{S}_i$, respectively. The information bottleneck losses are as the follow:

$$\begin{aligned}\mathcal{L}_{\text{UIB}}(\mathbf{S}_u^j, \mathbf{A}_u^j; \mathbf{Y}_{Rec}) &= -I(\mathbf{S}_u^j; \mathbf{Y}_{Rec}) + \beta \cdot I(\mathbf{A}_u^j; \mathbf{S}_u^j),\ j = 1, 2\\ \mathcal{L}_{\text{IIB}}(\mathbf{S}_i^j, \mathbf{A}_i^j; \mathbf{Y}_{Rec}) &= -I(\mathbf{S}_i^j; \mathbf{Y}_{Rec}) + \beta \cdot I(\mathbf{A}_i^j; \mathbf{S}_i^j),\ j = 1, 2\end{aligned} \tag{9}$$

where $I(\cdot;\cdot)$ is the mutual information function, and $\beta$ is a non-negative Lagrangian multiplier [32] employed to control the compression strength. $j$ denotes different meta-paths (Eq. 5). The $\mathbf{Y}_{Rec}$ is supervised signal [33], which in recommendation task corresponds to BPR [6] interaction pairs. However, $\mathbf{A}_u$ or $\mathbf{A}_i$ are not differentiable with respect to $\pi_{x,y}$, where $\pi_{x,y}$ denotes the probability of an edge existing between nodes $x$ and $y$ in $\mathbf{A}_u$ or $\mathbf{A}_i$. Next, we employ the reparameterization trick [31] to optimize the implicit graph structure $\mathbf{A}_u$ (or $\mathbf{A}_i$):

$$\begin{aligned}\mathbf{A}_u \text{ or } \mathbf{A}_i &= \bigcup_{x,y \in \boldsymbol{V}_u} \{a_{x,y} \sim \text{Ber}(\pi_{x,y})\},\\ \text{Ber}(\pi_{x,y}) &\approx \text{sigmoid}\left(\frac{1}{t}\left(\log\frac{\pi_{x,y}}{1-\pi_{x,y}} + \log\frac{\epsilon}{1-\epsilon}\right)\right),\end{aligned} \tag{10}$$

where $\epsilon$ follows a Uniform distribution $U(0, 1)$ and $t$ in $\mathbb{R}^+$ determines the temperature of the concrete distribution. We then implement a gradient-based autoencoder by setting thresholds $a_u$ and $a_i$ to filter the noise.

While directly computing mutual information (Eq. 9) is difficult, we complete this process by designing and optimizing the upper bound of $I(\mathbf{S}; \mathbf{A})$ and the lower bound of $(\mathbf{S}; \mathbf{Y}_{Rec})$.

- **Upper bound.** To eliminate unnecessary noise, we maximize the mutual information between the encoder and decoder through $I(\mathbf{A}_u; \mathbf{S}_u)$ and $I(\mathbf{A}_i; \mathbf{S}_i)$. The upper bound of the autoencoder's embeddings can be represented by the following method:

$$\begin{aligned}I(\mathbf{A}_u; \mathbf{S}_u) &\le \sum\sum p(\mathbf{S}_u) p(\mathbf{A}_u|\mathbf{S}_u) \log\frac{p(\mathbf{A}_u|\mathbf{S}_u)}{r(\mathbf{A}_u)},\\ I(\mathbf{A}_i; \mathbf{S}_i) &\le \sum\sum p(\mathbf{S}_i) p(\mathbf{A}_i|\mathbf{S}_i) \log\frac{p(\mathbf{A}_i|\mathbf{S}_i)}{r(\mathbf{A}_i)},\end{aligned} \tag{11}$$

  where $p(\cdot)$ is the probability distribution of the embeddings, and $p(A|B)$ denotes the conditional probability distribution. Moreover, $r(\cdot)$ is a Gaussian approximation of $p(\cdot)$.
- **Lower bound.** The lower bound of the learned autoencoder's embeddings, based on the supervision of the downstream recommendation task $\mathbf{Y}_{Rec}$, can be expressed as:

$$\begin{aligned}I(\mathbf{S}_u; \mathbf{Y}_{Rec}) &\ge \sum\sum p(\mathbf{Y}_{Rec}, \mathbf{S}_u) \log q(\mathbf{Y}_{Rec}|\mathbf{S}_u) + H(\mathbf{Y}_{Rec}),\\ I(\mathbf{S}_i; \mathbf{Y}_{Rec}) &\ge \sum\sum p(\mathbf{Y}_{Rec}, \mathbf{S}_i) \log q(\mathbf{Y}_{Rec}|\mathbf{S}_i) + H(\mathbf{Y}_{Rec}),\end{aligned} \tag{12}$$

  where $H(\cdot)$ is a setting that is irrelevant for optimization. To approximate $p(A, B)$, we utilize the non-negativity [32] of KL divergence to train $q(X|Y)$.

This is an example based on users to obtain the upper limit of $\mathcal{L}_{\text{UIB}}$, and similarly, we can obtain $\mathcal{L}_{\text{IIB}}$. To integrate Eq. 9 along with its upper and lower bounds, we formulate the objective as minimizing the following part:

$$\begin{aligned}\mathcal{L}_{\text{UIB}} &= -I\left(\mathbf{S}_u; \mathbf{Y}_{Rec}\right) + \beta \cdot I\left(\mathbf{S}_u; \mathbf{A}_u\right)\\ &\le -\sum\sum p\left(\mathbf{Y}_{Rec}, \mathbf{S}_u\right) \log q\left(\mathbf{Y}_{Rec} \mid \mathbf{S}_u\right)\\ &\quad + \beta \cdot \sum\sum p\left(\mathbf{A}_u\right) p\left(\mathbf{S}_u \mid \mathbf{A}_u\right) \log\frac{p\left(\mathbf{S}_u \mid \mathbf{A}_u\right)}{r\left(\mathbf{S}_u\right)},\end{aligned} \tag{13}$$

In order to approximate the upper limit mentioned above, we utilize the empirical distribution $p(\mathbf{Y}_{Rec}, \mathbf{A}_u) = p(\mathbf{A}_u)p(\mathbf{Y}_{Rec} \mid \mathbf{A}_u) = \frac{1}{\mathbf{N}}\sum_{n=1}^{\mathbf{N}} \delta\mathbf{Y}_n(\mathbf{Y}_{Rec})\delta\mathbf{A}_n(\mathbf{A}_u)$, and $\mathbf{N}$ is sampling number. The $\mathcal{L}_{\text{KL}}$ approximates the part mentioned in Eq. 13: $\mathcal{L}_{\text{UIB}} \le \mathcal{L}_{\text{KL}}$. The losses are as the follow:

$$\begin{aligned}\mathcal{L}_{\text{UIB}} &= \frac{1}{\mathbf{N}}\sum_{n=1}^{\mathbf{N}} -\log q(\mathbf{Y}_n \mid \mathbf{S}_u)\\ &\quad + \beta\frac{1}{\mathbf{N}}\sum_{n=1}^{\mathbf{N}} p(\mathbf{S}_u \mid \mathbf{A}_n) \log\frac{p(\mathbf{S}_u \mid \mathbf{A}_n)}{r(\mathbf{S}_u)},\end{aligned} \tag{14}$$

where a distribution $p(\mathbf{S}_u \mid \mathbf{A}_n)$ is $\mathcal{M}\left(\mathbf{S}_u \mid \mu\left(\mathbf{A}_n\right), \eta\left(\mathbf{A}_n\right)\right)$, and $\mathcal{M}$ is a Normal distribution. We leverage mean-pooing:

$$\left(\mu\left(\mathbf{A}_n\right), \eta\left(\mathbf{A}_n\right)\right) = \text{Pooling}\left(\left\{\mathbf{S}_u^1, \mathbf{S}_u^2, \mathbf{S}_u\right\}\right), \tag{15}$$

where $\mu(\cdot)$ and $\eta(\cdot)$ are calculated mean and standard deviation. The Eq. 15 represents the different user-based views, which $\mu\left(\mathbf{A}_n\right) \in \mathbb{R}^{M \times d/2}$ and $\eta\left(\mathbf{A}_n\right) \in \mathbb{R}^{M \times d/2}$. According to [22], [32], this derivation process can be proven. We utilize the same derivation to obtain the $\mathcal{L}_{\text{IIB}}$ based on items.

### C. Dual Contrastive Learning (DCL)

In this section, we propose Dual Contrastive Learning (DCL) to enhance the existing graph contrastive learning frameworks, which lack multi-meta-path augmentation. DCL aims to align the diverse intents of users and items at the meta-path level. The view level integrates these intents into real interactions, capturing global consistency at a higher level.

Following empirical practices [10], [11], we utilize the LightGCN for multi-layer iterations to obtain the user embeddings $\mathbf{E}_u^{'}$ and item embeddings $\mathbf{E}_i^{'}$, detailed in Eqs. 1 and 2. Next, we employ the obtained embeddings for DCL to achieve alignment between intent-intent contrast and intent-interaction contrast. Specifically, **User-item view:** $\mathbf{E}_u^{'}$ and $\mathbf{E}_i^{'}$; **Heterogeneous view:** $\mathbf{E}_{uu}^1$, $\mathbf{E}_{uu}^2$, $\mathbf{E}_{ii}^1$,$\mathbf{E}_{ii}^2$.

- **Intent-intent contrast (ICL).** The inputs for this part is the embeddings learned from the heterogeneous information view. We align the information of users and items

respectively. Inspired by [30], we adopt the InfoNCE [34] loss as the alignment objective and maximize the mutual information between the two. For user $u$, we represent each node in $\mathbf{E}_{uu}^{1}$ as $\mathbf{e}^{'}$, and each node in $\mathbf{E}_{uu}^{2}$ as $\mathbf{e}^{''}$. The defined intent-intent contrast loss (ICL) is as follows:

$$\mathcal{L}_{\text{ICL}}^{\text{user}} = \sum_{i \in \mathcal{B}} -\log \frac{\exp\left(s\left(\mathbf{e}_i^{'}, \mathbf{e}_i^{''}\right)/\tau^{'}\right)}{\sum_{j \in \mathcal{B}} \exp\left(s\left(\mathbf{e}_i^{'}, \mathbf{e}_j^{''}\right)/\tau^{'}\right)}, \tag{16}$$

where $i$, $j$ are users/items in a sampled batch $\mathcal{B}$. $s(\cdot,\cdot)$ means the cosine similarity, and $\tau^{'}$ is a temperature coefficient. Following the same approach, we obtain the contrastive loss $\mathcal{L}_{\text{ICL}}^{\text{item}}$ of items. Meanwhile, some studies [33], [35] indicate that minimizing the InfoNCE loss corresponds to increasing the mutual information. We utilize positive and negative samples obtained from sampled batch $\mathcal{B}$ to calculate the mutual information between different meta-paths on the user side (or item side). Intent-intent contrast aims to align specific intents (meta-paths) and capture similarities between different intents.

- **Intent-interaction contrast (IICL).** The inputs for this part consists of embeddings learned from combining user-item and heterogeneous views, denoted as: $\left(\mathbf{E}_u^{'} + \mathbf{E}_{uu}^{1}\right)$ and $\left(\mathbf{E}_u^{'} + \mathbf{E}_{uu}^{2}\right)$. For user $u$, we denote each nodes by $\mathbf{z}^{'}$ and $\mathbf{z}^{''}$. Considering the user preferences and the item relationships, we model these intents using heterogeneous information. Intent refers to the motivation behind a user's choice of items, and this process is often traceable. For example, two users who happen to belong to the same age group and occupation are empirically more likely to interact with similar items. To learn the intent representations behind the views enhanced by multiple meta-paths, we use the same loss function to further align specific users (or items):

$$\mathcal{L}_{\text{IICL}}^{\text{user}} = \sum_{i \in \mathcal{B}} -\log \frac{\exp\left(s\left(\mathbf{z}_i^{'}, \mathbf{z}_i^{''}\right)/\tau^{''}\right)}{\sum_{j \in \mathcal{B}} \exp\left(s\left(\mathbf{z}_i^{'}, \mathbf{z}_j^{''}\right)/\tau^{''}\right)}, \tag{17}$$

where $\tau^{''}$ is the temperature coefficient for intent-interaction contrast. Based on the similarly enhanced embeddings $\left(\mathbf{E}_i^{'} + \mathbf{E}_{ii}^{1}\right)$ and $\left(\mathbf{E}_i^{'} + \mathbf{E}_{ii}^{2}\right)$, we can obtain the loss $\mathcal{L}_{\text{IICL}}^{\text{item}}$ for items.

### D. Model Optimization

To improve self-supervised recommendation, we utilize a multi-task joint learning to formulate the final optimization objective. First, we organize the loss functions of the main modules: BAE and DCL. BAE employs the information bottleneck principle to denoise in the autoencoder, from which we can obtain:

$$\mathcal{L}_{\text{IB}} = \mathcal{L}_{\text{UIB}} + \mathcal{L}_{\text{IIB}}, \tag{18}$$

Since the users' and items' views are symmetrical, we do not set separate coefficients for them. However, DCL has a hierarchical structure, which we summarize as follows:

$$\begin{aligned}\mathcal{L}_{\text{DCL}} &= \lambda_{\text{ICL}} \cdot \mathcal{L}_{\text{ICL}} + \lambda_{\text{IICL}} \cdot \mathcal{L}_{\text{IICL}} \\ &= \lambda_{\text{ICL}} \cdot \left(\mathcal{L}_{\text{ICL}}^{\text{user}} + \mathcal{L}_{\text{ICL}}^{\text{item}}\right) + \lambda_{\text{IICL}} \cdot \left(\mathcal{L}_{\text{IICL}}^{\text{user}} + \mathcal{L}_{\text{IICL}}^{\text{item}}\right),\end{aligned} \tag{19}$$

where $\lambda_{\text{ICL}}$ and $\lambda_{\text{IICL}}$ represent the weights for intent-intent contrast and intent-interaction contrast, respectively. Next, after summarizing the loss functions of the modules we proposed, we introduce the pairwise Bayesian Personalized Ranking (BPR) loss [6] used in recommendation task:

$$\mathcal{L}_{\text{BPR}} = -\frac{1}{|\mathcal{B}|} \sum_{(i,j,k) \in B} \log \sigma\left(\mathbf{d}_i^{\top}\mathbf{d}_j - \mathbf{d}_i^{\top}\mathbf{d}_k\right), \tag{20}$$

where $\mathcal{B} = \{(i,j,k) \mid A_{i,j} = 1, A_{i,k} = 0\}$ is the training data, and embeddings $\mathbf{d} \in \left\{\mathbf{E}_u^{'} + \mathbf{E}_{uu}^{1} + \mathbf{E}_{uu}^{2}\right\}$ are obtained by weighting the main task and heterogeneous information. This loss function enforces that the predicted scores for observed interactions are higher than those for unobserved interactions. Finally, the complete optimization objective of IHGCL is as the follow:

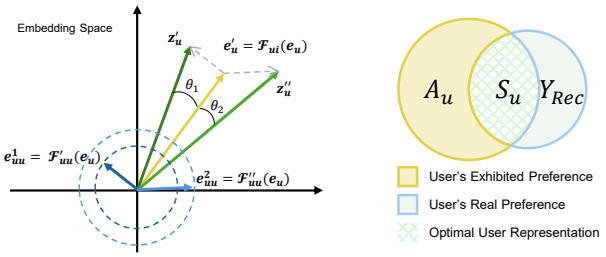
$$\mathcal{L}_{\text{IHGCL}} = \mathcal{L}_{\text{BPR}} + \lambda_1 \cdot \mathcal{L}_{\text{IB}} + \mathcal{L}_{\text{DCL}} + \lambda_2 \cdot \|\Theta\|_2^2 \tag{21}$$

where $\lambda_1$ and $\lambda_2$ are adjustable weights, and $\|\Theta\|_2^2$ are trainable model parameters and $L_2$ regularization. Because $\lambda_{\text{ICL}}$ and $\lambda_{\text{IICL}}$ have already determined the weight of $\mathcal{L}_{\text{DCL}}$, there is no need to set them again here. Algorithm 1 provides the overall process of the learning steps for IHGCL.

**Algorithm 1** The IHGCL Learning Algorithm

**Input:** User-item interaction matrix $\mathbf{A}$, meta-path-based subgraph $\mathbf{A}_u^1$, $\mathbf{A}_u^2$, $\mathbf{A}_i^1$, $\mathbf{A}_i^2$, maximum training epochs $E$ and parameters required for the training process
**Output:** Trained node embeddings

```
Initialize all parameters;
for epoch in {1, 2, ..., E} do
    Obtain high-order embedding E'_u and E'_i of the input graph A via l-layer LightGCN, applying Eq. 2;
    Obtain intent embeddings E^1_uu, E^2_uu, E^1_ii, E^2_ii using the BAE module via meta-path-based subgraph A^1_u, A^2_u, A^1_i, A^2_i, applying Eqs. 5–8;
    Learn distributions of μ(A_n), η(A_n) via pooling operation on S^1_u, S^2_u via Eq. 15;
    Optimize IB via loss L_IB according to Eq. 14;
    Perform DCL augmentation based on two views via loss L_ICL and L_IICL according to Eqs. 16 and 17;
    Optimize BPR loss L_BPR via parameter regularizer via Eq. 20;
    Joint optimization of IHGCL following Eq. 21;
end for
return all parameters and user and item embeddings;
```




(a) DCL (b) BAE

Fig. 5: Theoretical analysis of DCL and BAE.

## IV. Model Analysis

### A. Theoretical Analysis

In this section, we perform an in-depth analysis of IHGCL to answer how recommendation tasks can benefit from heterogeneous information. We decompose the model into the two proposed components: DCL and BAE.

**DCL:** We utilize heterogeneous information to model the multi-view intent of users and items for the main task augmentation. This process is formally described as follows: Given a user node $u$ and its initialized representation $\mathbf{e}_u$ in the $d$-dimensional embedding space (Eq. 17):

$$\begin{aligned}\mathbf{z}'_u &= \mathbf{e}'_u + \mathbf{e}^1_{uu} = \mathcal{F}_{ui}(\mathbf{e}_u) + \mathcal{F}'_{uu}(\mathbf{e}_u),\\ \mathbf{z}''_u &= \mathbf{e}'_u + \mathbf{e}^2_{uu} = \mathcal{F}_{ui}(\mathbf{e}_u) + \mathcal{F}''_{uu}(\mathbf{e}_u),\end{aligned} \tag{22}$$

where $\mathcal{F}'_{uu}$ and $\mathcal{F}''_{uu}$ are BAE, and $\mathcal{F}_{ui}$ is LightGCN encoder.

As illustrated in Figure 5 (a), we assume that in the contrastive loss, $\mathcal{F}_{ui}$ encodes the multi-hop information of user-item interactions, while $\mathcal{F}_{uu}$ models the preferences of users contained within different meta-paths in heterogeneous information. These preferences (blue) interact with embeddings in the interaction domain (yellow), leading in corresponding deviations ($\theta_1$ and $\theta_2$) to produce enhanced embeddings. Notably, the process of contrastive learning, which aligns and integrates embeddings with various intents, aids the model in better perceiving the intents of users and items. This point will be validated in Section V-C. Therefore, having high-quality user and item intents is crucial for effective contrastive learning, which explains the necessity of introducing BAE.

**BAE:** We combine the techniques of masked autoencoder and information bottleneck (IB) for denoising, aiming to learn embeddings of users or items devoid of irrelevant information. The mutual information in Figure 5 (b) can be expressed as:

$$\mathcal{L}_{\text{IB}}(\mathbf{S}_u, \mathbf{A}_u; \mathbf{Y}_{Rec}) = -I(\mathbf{S}_u; \mathbf{Y}_{Rec}) + \beta \cdot I(\mathbf{A}_u; \mathbf{S}_u), \tag{23}$$

where $\mathbf{S}_u$ denotes the weighted embeddings of the encoder and decoder, which are the user preferences we need to optimize. The connections brought by the meta-paths merely represent the users' exhibited preferences ($\mathbf{A}_u$), which often contain noise, while $\mathbf{Y}_{Rec}$ is the downstream recommendation information. The IB strategy encourages the representation to capture the minimal sufficient information needed for downstream tasks, which involves maximizing the intersection between $\mathbf{S}_u$ and $\mathbf{Y}_{Rec}$, and minimizing the intersection between $\mathbf{S}_u$ and $\mathbf{A}_u$. Specifically, Eqs. 9 to 15 illustrate the denoised process for heterogeneous information.

TABLE I: Statistics of datasets used in this paper.

| Dataset | User # | Item # | Interaction # | Meta-paths |
|---|---|---|---|---|
| Last.fm | 1,892 | 17,632 | 92,834 | UU, UATAU, AA, ATA |
| Amazon | 6,170 | 2,753 | 195,791 | UIU, UIBIU, IBI, ICI |
| Yelp | 16,239 | 14,284 | 198,397 | UU, UBU, BCiB, BCaB |
| Douban Book | 13,024 | 22,347 | 792,062 | UU, UGU, BAB, BYB |
| Movielens-1M | 6040 | 3952 | 1,000,209 | UMU, UOU, MUM, MGM |
| Douban Movie | 13,367 | 12,677 | 1,068,278 | UU, UGU, MAM, MDM |

### B. Time Complexity Analysis

This section discusses the batch time complexity [11]. The user-item view's time complexity is $\mathcal{O}(2|\mathcal{V}_{ui}| \cdot Ld)$, where $|\mathcal{V}_{ui}|$ denotes the edge number of a interaction graph $\mathbf{R}$. $L$ is the GCN layers and $d$ is the feature dimension. For the heterogeneous view, IHGCL has four meta-path-based subgraphs: $G^1_U, G^2_U, G^1_I, G^2_I$ (Eq. 5). So, the time complexity of the heterogeneous view is $\mathcal{O}\left(2|\boldsymbol{V}_u| \cdot L'd + 2|\boldsymbol{V}_i| \cdot L'd\right)$, where $L'$ is the number of autoencoder layers $(L_E + L_D)$. Then, the self-supervised loss contains $\lambda_{\text{IB}}$ and $\lambda_{\text{DCL}}$. The time complexity of $\lambda_{\text{IB}}$ primarily depends on the number of nodes and can generally be considered as $\mathcal{O}(Md + Nd)$ in most cases. To calculate the complexity of $\lambda_{\text{DCL}}$, we get doubled time $\mathcal{O}(2Bd + 2BCd)$ compared to general contrastive learning, where $C$ represents the node number in a batch and $B$ denote the batch size. In the recommendation task, the complexity of the additional $\mathcal{L}_{\text{BPR}}$ is $\mathcal{O}(2Bd)$. Thus, the overall time complexity of encoding is $\mathcal{O}\left(2(|\mathcal{V}_{ui}| \cdot L + |\boldsymbol{V}_u| \cdot L' + |\boldsymbol{V}_i| \cdot L')d\right)$, and losses complexity are $\mathcal{O}((M + N + 4B + 2BC)d)$. This complexity is the time complexity for each epoch during training, and the number of training epochs determine the overall training time.

## V. Experiment

In this section, we conduct extensive experiments and answer the following research questions:

- **RQ1:** How does IHGCL compare to the current state-of-the-art (SOTA) models in terms of performance?
- **RQ2:** What are the reasons for IHGCL's superior performance, and how does it differ from existing models?
- **RQ3:** What impact does the selection of meta-paths have on the model?
- **RQ4:** Are the key components in our IHGCL delivering the expected performance gains?
- **RQ5:** How do different hyperparameters affect IHGCL?

### A. Experiment Setup

*1) Datasets:* The performance of IHGCL has been validated on six real-world datasets. A statistical overview of all the datasets is presented in Table I. We utilize various datasets for recommender systems: **Last.fm**[1] for music, **Amazon** [23] for products, **Yelp** [24] for businesses, **Douban** for books [23] and movies[2] and **Movielens-1M**[3] for movies, predicting user interactions with diverse entities like artists, items, businesses, books and movies.

[1]https://www.last.fm/
[2]https://m.douban.com/
[3]https://grouplens.org/datasets/movielens/

TABLE II: Performance Comparison of Different Recommendation Methods with Varying Top-N on Six Datasets. The best results are shown in bold and the second-best results are underlined. R@K represents Recall@K, and N@K represents NDCG@K, and the improvement is significant based on two-tailed paired t-test.

| Datasets | Metrics | Graph Recommendation Models | | | | Intent Recommendation Models | | | Heterogeneous Graph Recommendation Models | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | LightGCN | SGL | SimGCL | LightGCL | DisenHAN | DCCF | BIGCF | HERec | HAN | HeCo | SMIN | HGCL | IHGCL | Improv. | $p$-value |
| Last.fm | R@5 | 0.1242 | 0.1293 | 0.1335 | 0.1302 | 0.1158 | 0.1306 | 0.1347 | 0.1117 | 0.1084 | 0.1169 | 0.1231 | 0.1298 | **0.1370** | 1.71% | $4.7e^{-5}$ |
| | N@5 | 0.2673 | 0.2786 | 0.2859 | 0.2805 | 0.2581 | 0.2804 | 0.2852 | 0.2386 | 0.2291 | 0.2565 | 0.2671 | 0.2777 | **0.2913** | 1.89% | $9.2e^{-4}$ |
| | R@10 | 0.1866 | 0.1923 | 0.1966 | 0.1938 | 0.1772 | 0.1945 | 0.1959 | 0.1704 | 0.1678 | 0.1757 | 0.1818 | 0.1907 | **0.2044** | 3.97% | $3.3e^{-3}$ |
| | N@10 | 0.2356 | 0.2404 | 0.2467 | 0.2419 | 0.2234 | 0.2429 | 0.2476 | 0.2093 | 0.2025 | 0.2226 | 0.2299 | 0.2390 | **0.2542** | 2.67% | $6.5e^{-6}$ |
| | R@20 | 0.2626 | 0.2714 | 0.2760 | 0.2730 | 0.2516 | 0.2734 | 0.2752 | 0.2468 | 0.2382 | 0.2514 | 0.2581 | 0.2698 | **0.2824** | 2.32% | $8.8e^{-4}$ |
| | N@20 | 0.2628 | 0.2694 | 0.2755 | 0.2711 | 0.2492 | 0.2722 | 0.2764 | 0.2385 | 0.2284 | 0.2497 | 0.2574 | 0.2683 | **0.2815** | 1.85% | $2.4e^{-8}$ |
| Amazon | R@5 | 0.0653 | 0.0704 | 0.0742 | 0.0709 | 0.0608 | 0.0718 | 0.0743 | 0.0571 | 0.0546 | 0.0618 | 0.0640 | 0.0706 | **0.0762** | 2.56% | $5.2e^{-7}$ |
| | N@5 | 0.0875 | 0.0948 | 0.1011 | 0.0974 | 0.0821 | 0.0990 | 0.1014 | 0.0766 | 0.0748 | 0.0840 | 0.0873 | 0.0978 | **0.1040** | 2.56% | $5.8e^{-4}$ |
| | R@10 | 0.1028 | 0.1084 | 0.1197 | 0.1038 | 0.0958 | 0.1147 | 0.1193 | 0.0903 | 0.0885 | 0.0995 | 0.1031 | 0.1094 | **0.1230** | 2.76% | $3.2e^{-6}$ |
| | N@10 | 0.0976 | 0.1041 | 0.1138 | 0.0987 | 0.0905 | 0.1061 | 0.1127 | 0.0853 | 0.0832 | 0.0934 | 0.0969 | 0.1062 | **0.1161** | 2.02% | $7.8e^{-6}$ |
| | R@20 | 0.1592 | 0.1646 | 0.1751 | 0.1574 | 0.1508 | 0.1705 | 0.1758 | 0.1420 | 0.1385 | 0.1519 | 0.1569 | 0.1670 | **0.1805** | 2.67% | $8.2e^{-8}$ |
| | N@20 | 0.1151 | 0.1215 | 0.1306 | 0.1140 | 0.1098 | 0.1255 | 0.1307 | 0.1017 | 0.0988 | 0.1097 | 0.1135 | 0.1247 | **0.1346** | 2.98% | $5.3e^{-5}$ |
| Yelp | R@5 | 0.0350 | 0.0371 | 0.0398 | 0.0373 | 0.0317 | 0.0379 | 0.0405 | 0.0282 | 0.0264 | 0.0318 | 0.0336 | 0.0367 | **0.0425** | 4.94% | $4.7e^{-6}$ |
| | N@5 | 0.0406 | 0.0414 | 0.0441 | 0.0421 | 0.0349 | 0.0427 | 0.0446 | 0.0330 | 0.0316 | 0.0352 | 0.0352 | 0.0400 | **0.0473** | 6.05% | $6.4e^{-5}$ |
| | R@10 | 0.0584 | 0.0622 | 0.0661 | 0.0631 | 0.0519 | 0.0632 | 0.0658 | 0.0460 | 0.0423 | 0.0523 | 0.0540 | 0.0628 | **0.0693** | 4.84% | $2.8e^{-4}$ |
| | N@10 | 0.0471 | 0.0509 | 0.0525 | 0.0512 | 0.0409 | 0.0504 | 0.0519 | 0.0378 | 0.0357 | 0.0412 | 0.0409 | 0.0512 | **0.0546** | 4.00% | $4.0e^{-5}$ |
| | R@20 | 0.0883 | 0.0961 | 0.1003 | 0.0980 | 0.0820 | 0.0973 | 0.1009 | 0.0766 | 0.0734 | 0.0823 | 0.0854 | 0.0963 | **0.1044** | 3.47% | $5.1e^{-7}$ |
| | N@20 | 0.0555 | 0.0604 | 0.0616 | 0.0608 | 0.0500 | 0.0605 | 0.0620 | 0.0469 | 0.0440 | 0.0497 | 0.0500 | 0.0605 | **0.0649** | 4.68% | $2.9e^{-6}$ |
| Douban Book | R@5 | 0.0626 | 0.0728 | 0.0752 | 0.0726 | 0.0567 | 0.0725 | 0.0742 | 0.0509 | 0.0480 | 0.0564 | 0.0597 | 0.0708 | **0.0800** | 6.38% | $6.3e^{-6}$ |
| | N@5 | 0.1353 | 0.1547 | 0.1560 | 0.1534 | 0.1274 | 0.1541 | 0.1551 | 0.1027 | 0.0984 | 0.1284 | 0.1330 | 0.1509 | **0.1616** | 3.59% | $7.7e^{-5}$ |
| | R@10 | 0.0968 | 0.1136 | 0.1147 | 0.1130 | 0.0909 | 0.1137 | 0.1149 | 0.0836 | 0.0801 | 0.0905 | 0.0934 | 0.1103 | **0.1190** | 3.57% | $5.8e^{-4}$ |
| | N@10 | 0.1312 | 0.1507 | 0.1519 | 0.1495 | 0.1245 | 0.1506 | 0.1535 | 0.1042 | 0.0975 | 0.1248 | 0.1298 | 0.1470 | **0.1556** | 1.37% | $5.1e^{-4}$ |
| | R@20 | 0.1471 | 0.1665 | 0.1701 | 0.1666 | 0.1384 | 0.1665 | 0.1679 | 0.1312 | 0.1255 | 0.1387 | 0.1433 | 0.1627 | **0.1726** | 1.47% | $3.2e^{-7}$ |
| | N@20 | 0.1370 | 0.1551 | 0.1579 | 0.1556 | 0.1311 | 0.1556 | 0.1591 | 0.1130 | 0.1076 | 0.1299 | 0.1339 | 0.1547 | **0.1611** | 1.26% | $3.8e^{-5}$ |
| Movielens | R@5 | 0.0960 | 0.0992 | 0.1042 | 0.1002 | 0.0912 | 0.1006 | 0.1048 | 0.0871 | 0.0813 | 0.0912 | 0.0930 | 0.0985 | **0.1071** | 2.20% | $2.5e^{-6}$ |
| | N@5 | 0.4180 | 0.4235 | 0.4366 | 0.4281 | 0.4102 | 0.4307 | 0.4376 | 0.3873 | 0.3865 | 0.4124 | 0.4150 | 0.4256 | **0.4482** | 2.42% | $4.4e^{-5}$ |
| | R@10 | 0.1593 | 0.1648 | 0.1721 | 0.1654 | 0.1505 | 0.1667 | 0.1680 | 0.1444 | 0.1331 | 0.1516 | 0.1521 | 0.1633 | **0.1764** | 2.50% | $6.0e^{-4}$ |
| | N@10 | 0.3903 | 0.3999 | 0.4087 | 0.4035 | 0.3811 | 0.4034 | 0.4050 | 0.3613 | 0.3550 | 0.3824 | 0.3828 | 0.4075 | **0.4172** | 2.08% | $5.3e^{-7}$ |
| | R@20 | 0.2509 | 0.2569 | 0.2667 | 0.2578 | 0.2355 | 0.2606 | 0.2625 | 0.2303 | 0.2127 | 0.2374 | 0.2378 | 0.2530 | **0.2713** | 1.73% | $6.2e^{-6}$ |
| | N@20 | 0.3763 | 0.3837 | 0.3942 | 0.3863 | 0.3649 | 0.3869 | 0.3916 | 0.3494 | 0.3376 | 0.3664 | 0.3666 | 0.3843 | **0.4010** | 1.72% | $8.1e^{-7}$ |
| Douban Movie | R@5 | 0.0700 | 0.0764 | 0.0818 | 0.0774 | 0.0693 | 0.0786 | 0.0816 | 0.0683 | 0.0663 | 0.0692 | 0.0695 | 0.0752 | **0.0847** | 3.55% | $4.0e^{-3}$ |
| | N@5 | 0.2085 | 0.2098 | 0.2174 | 0.2111 | 0.2050 | 0.2103 | 0.2153 | 0.1778 | 0.1710 | 0.2064 | 0.1895 | 0.2099 | **0.2265** | 4.19% | $9.5e^{-5}$ |
| | R@10 | 0.1139 | 0.1202 | 0.1258 | 0.1220 | 0.1114 | 0.1223 | 0.1253 | 0.1102 | 0.1018 | 0.1110 | 0.1123 | 0.1180 | **0.1274** | 1.27% | $5.5e^{-4}$ |
| | N@10 | 0.2007 | 0.2025 | 0.2093 | 0.2036 | 0.1981 | 0.2051 | 0.2108 | 0.1764 | 0.1719 | 0.1974 | 0.1838 | 0.2021 | **0.2253** | 6.88% | $4.6e^{-5}$ |
| | R@20 | 0.1781 | 0.1880 | 0.1972 | 0.1893 | 0.1720 | 0.1904 | 0.1963 | 0.1757 | 0.1692 | 0.1736 | 0.1743 | 0.1849 | **0.2003** | 1.57% | $1.8e^{-7}$ |
| | N@20 | 0.2001 | 0.2046 | 0.2158 | 0.2052 | 0.1978 | 0.2084 | 0.2144 | 0.1828 | 0.1774 | 0.1961 | 0.1860 | 0.2037 | **0.2195** | 1.72% | $9.5e^{-5}$ |

*2) Baselines:* We respectively select the most representative baseline model for comparison. **ID-GRec**[4] provides baseline implementations for intent-based models and general recommendation models. **HGNN-baselines**[5] includes heterogeneous graph neural network models.

**Graph Recommendation Models.**

- **LightGCN** [5] removes the feature transformation and nonlinear activations of GCN to achieve light recommendation.
- **SGL** [10] generates contrast views by edge dropout to aid contrastive learning to enhance recommendation.
- **SimGCL** [11] considers the relationship between neighbor nodes to enhance collaborative filtering.
- **LightGCL** [9] uses singular value decomposition (SVD) to construct lightweight contrastive views.

**Intent Recommendation Models.**

- **DisenHAN** [17] employs meta relation decomposition and disentangled propagation layers to capture semantics.
- **DCCF** [12] enhances self-supervised signals by learning disentangled representations with global context.
- **BIGCF** [27] explores the individuality and collectivity of intents behind interactions for collaborative filtering.

**Heterogeneous Graph Recommendation Models.**

- **HERec** [23] integrates meta-path-based random walks for node embeddings in HINs with fusion functions.
- **HAN** [19] employs node-level and semantic-level attention to capture node and meta-path importance.
- **HeCo** [20] uses collaborative supervised contrast between network schema view and meta-path view.
- **SMIN** [36] utilizes metagraph informax network to enhance user preference representation for social recommendation.
- **HGCL** [15] integrates heterogeneous relational semantics by leveraging contrastive self-supervised learning.

*3) Implementations:* To ensure a fair comparison, we follow the sampling method and dataset format of classic works [5], [10], [37]. All models (including baselines) are retrained until convergence using the Adam optimizer and Xavier [38] initialization for embeddings. For some classic heterogeneous graph neural network models, such as HAN [19] and HeCo [20], we replace the supervised loss with the BPR [6] loss. We opted for LightGCN [5] as the encoder due to its consistent ability to balance optimal performance and computational efficiency across diverse datasets. We tune the parameters for the baselines on each dataset to ensure they achieve optimal performance. For general settings, GCN-based models maintain consistent layer numbers, learning rate 0.001, embedding size 64, and fixed batch size 4096. For the unique parameters of IHGCL, we analyze them in RQ4.

### *B. Overall Performance (RQ1)*

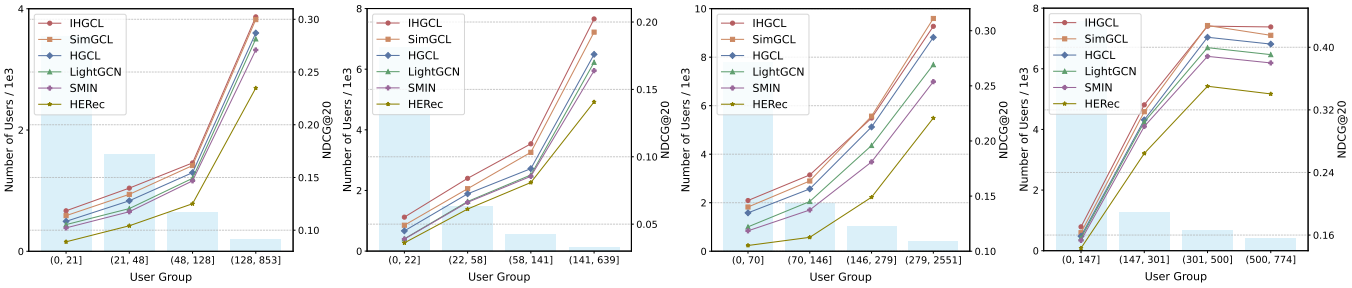
Table II reports the recommendation performance of all baseline models on six public datasets, and we summarize possible observations or explanations for these outcomes.

[4]https://github.com/BlueGhostYi/ID-GRec

[5]https://github.com/wangyu0627/HGNN-baselines

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3536096






(a) Amazon (b) Yelp (c) Douban Book (d) Douban Movie

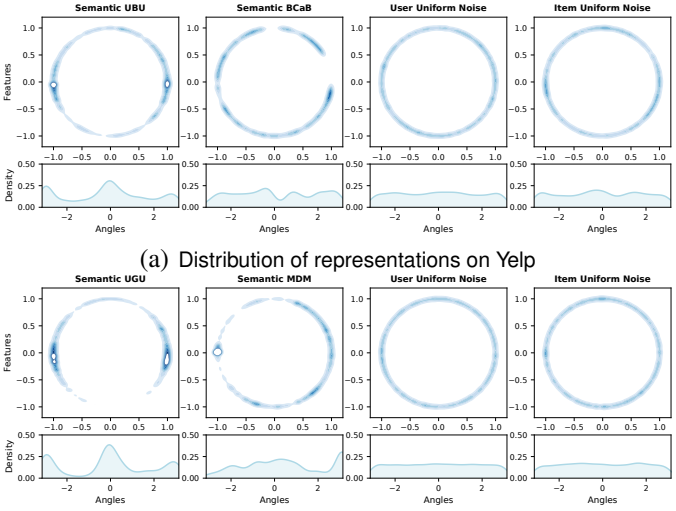
Fig. 6: Performance comparison of different sparsity levels. The bar graph shows users' number per group on the left y-axis, and the line graph shows the performance of each method w.r.t. NDCG@20 on the right y-axis, with the x-axis denoting the interval of interactions per user group.

- IHGCL demonstrates the best performance across all metrics on all datasets. Quantitatively, compared to the current SOTA baselines of HG-based recommendation, we achieved a significant improvement (douban movie for 11.48% and amazon for 12.43%). These experimental results prove the superiority and rationality of the proposed IHGCL.
- IHGCL achieves substantial advancements over the models based on HGNN (e.g., HGCL [15] and SMIN [36]). This progress demonstrates the necessity of constructing multiple views of users and items through heterogeneous information for contrast. The performance degradation caused by noise in meta-paths indicates that HG-based models are insufficient in handling noise. Our BAE effectively mitigates this noise, while other methods lack a denoising mechanism.
- IHGCL exhibits a distinct performance advantage over general recommender systems (e.g., SGL [10] and SimGCL [11]). We have overcome the significant noise issues inherent in heterogeneous information and effectively utilized this information to extract user and item intents. The experimental results prove that IHGCL can better mine user preferences for enhancing CL-based recommendation.
- Considering the effectiveness in intent modeling, we compared the latest works in heterogeneous graph and bipartite graph scenarios. It can be observed that BIGCF [27] achieved better results than SimGCL on multiple metrics, proving that personalized intent recommendations can help users find what they truly need. Meanwhile, IHGCL uses meta-paths as intent-guided links to connect users or items, enhancing the model's ability to capture intents.

### C. Explainability and Visualization (RQ2)

In this section, we illustrate how intent enhances our CL-based recommendation, and experiments demonstrate that it provides more personalized recommendations for users with sparse interactions.

*1)* ***Comparisons w.r.t. Data Sparsity****:* Existing models often struggle to provide reasonable recommendation for users with fewer interactions, which is one of the critical factors hindering performance. To verify IHGCL's ability to explore the intents of these users, we conducted sparsity tests on four datasets in Figure 6. Following the settings [7] and [28], we divided users into four groups based on the scale of their interaction, ensuring that the total number of interactions in each group was roughly equal. Taking the sparsest dataset 'Yelp' as an example, the first group contains 6,721 users with interactions that do not exceed 22. In other words, 75% of users in the test set exhibit very sparse interactions. Figure 6 demonstrates that as the interaction scale increases, the performance of all methods improves significantly, highlighting that more interaction data is crucial. IHGCL achieved performance improvements in the sparsest user group by 4.21%, 10.19%, 4.60%, and 4.05% (relative to SimGCL [11]), proving that modeling heterogeneous information to capture the intents of users and items is crucial for alleviating sparsity issue.




(a) Distribution of representations on Yelp

(b) Distribution of representations on Douban Movie

Fig. 7: IHGCL's intents augmentation and SimGCL's uniform noise. We plot feature distributions with Gaussian kernel density estimation (KDE) [39] (the darker the color is, the more points fall in that area.) and KDE on angles.

*2)* ***Modeling Heterogeneous Information****:* According to Eq. 22, we utilize t-SNE [40] to visualize the intents of the modeled users and items. As shown in Figure 7, following the setting [11], we input the trained embeddings into BAE to obtain intent-based representations of users and items. For example, in the Yelp dataset, the two columns on the left represent features obtained by encoding intent behind meta-paths, while the two columns on the right represent uniform noise added at each layer by SimGCL. We observe that, compared to the uniform noise in SimGCL, the users and items in IHGCL exhibit an apparent deviation. These deviation guide

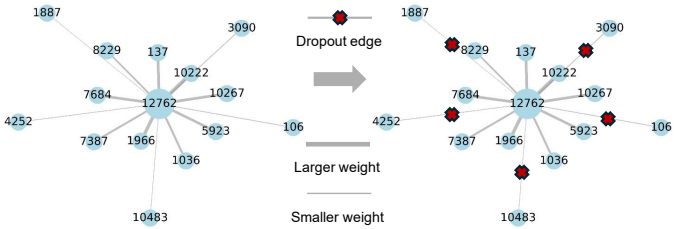



Fig. 8: Meta-path UU on Douban Book. The left shows the original interaction graph of user 12762 before training, while the right is graph constrained by the information bottleneck.

TABLE III: Performance w.r.t. Meta-path Number.

| Meta-path | Douban Book | | Yelp | | Movielens | |
|---|---|---|---|---|---|---|
| Number | R@20 | N@20 | R@20 | N@20 | R@20 | N@20 |
| **2-MP** | 0.1726 | 0.1611 | 0.1044 | 0.0649 | **0.2713** | **0.4010** |
| **3-MP** | **0.1734** | **0.1620** | 0.1049 | 0.0653 | 0.2700 | 0.3992 |
| **4-MP** | 0.1721 | 0.1606 | **0.1050** | **0.0655** | 0.2688 | 0.3985 |

the CL-based gradient, leading to parameter updates in the intended direction. We explain this phenomenon in Figure 5 (a) and utilize these preferences to improve the representations of users and items. Such results demonstrate that modeling heterogeneous information facilitates the understanding of the intents of users and items, as evidenced by the sparsity test.

*3)* ***Case Study of BAE****:* As shown the left of Figure 8, before training begins, user 12762 is connected to other users through the meta-path UU. After training, we obtain the re-parameterized edge weights by one iteration of BAE using the user embeddings. The thickness of the edges in the graph represents the magnitude of these weights. On the right side, BAE reconstructs the graph based on a threshold ($cf$.Eq. 10), removing some connections to achieve a minimized graph based on the information bottleneck principle. Overall, the superior recommendation performance and the graph structure minimization process demonstrate that BAE can effectively mitigate noise issues, addressing **CH2**.

### *D. In-depth Analysis of Meta-paths (RQ3)*

In Section II, we select two user and item subgraphs based on meta-paths for the model. To study the effect of incorporating multiple intents into our model, we also perform experiments to explore the following two questions:

*1)* ***Selection Impact with Two Meta-paths****:* CL-based recommendation methods typically construct two augmented views for alignment to mine consistency information. We follow this paradigm [10], [11], so in the experiments, we select two intents for users and items, respectively. In Figure 9, we examine the impact of different meta-path combinations on model performance across four datasets. For example, in the Movielens dataset, the second brown bar labeled 'UMU&UAU' indicates the replacement of the path 'UOU' in the optimal combination with 'UAU'. We ranked these combinations in descending order and observed that: i) The meta-path UU (i.e., social intent) compared with other intents often achieves better recommendation accuracy. We believe this is due to the positive role of social context and network homogeneity in shaping user influence. ii) From the coordinate axis, it can be seen that the selection of intents can be diverse and does not significantly impact the model's performance. The lowest values across all datasets are still better than the baseline in Table II, indicating that IHGCL can adaptively capture useful information during the alignment of intents.

*2)* ***Generalization Ability with More Meta-paths****:* To explore the impact of aligning more intents on recommendation accuracy, we compared multiple views pairwise to generate more supervision signals. As shown in Table III, we designed three variants of the model on three datasets: 2-MP is the existing model, 3-MP adds one user and one item meta-path to the original, and so on. It can be observed that with the increase in contrastive views (i.e., the addition of more meta-paths), the model performance does not significantly improve, and even declined. We believe the information in these views is limited, and over-mining may lead to an information bottleneck [33]. That is, the useful information that can be provided by the views has already been fully utilized. Adding more views or adjusting the model structure is unlikely to yield more useful information, thus limiting performance improvement. Moreover, this slight increase leads to a significant rise in complexity [41], hence we do not adopt this strategy.

### *E. Ablation Study (RQ4)*

In this section, we validate the effectiveness of the key components in our IHGCL and provide possible explanations regarding the results. Here is our explanation of variants for several models.

- IHGCL$_{\text{w/o DCL}}$: remove the dual contrastive learning;
- IHGCL$_{\text{w/o BAE}}$: remove the bottlenecked autoencoder, and use a GCN [5] encoder directly recommendation;
- IHGCL$_{\text{w/o IB}}$: remove the information bottleneck in BAE;
- IHGCL$_{\text{w/o ICL}}$: remove the intent-intent contrastive learning in DCL but keep intent-interaction contrastive learning;
- IHGCL$_{\text{w/o IICL}}$: remove the intent-interaction contrastive learning in DCL.

Table IV shows the experimental results of all variants on all datasets, and we have the following findings:

The experimental results confirm that the DCL module significantly contributes to improving recommendation performance, emphasizing the necessity of leveraging heterogeneous information to model user and item intents from a contrastive learning perspective. In contrast, the BAE module holds secondary importance, as it effectively alleviates noise issues by helping the model establish reliable preferences and capture semantic connections.

Subsequently, we further dissect the DCL into ICL and IICL to investigate their roles. It is noteworthy that IICL serves as the principal loss function. This suggests that intent-interaction contrast through aligned heterogeneous information significantly enhances intents, while intent-intent contrast regulates intent uniformity. However, an anomaly is that the experimental results of IICL on the Last.fm and Douban Book datasets are lower than those of DCL. We believe that

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3536096



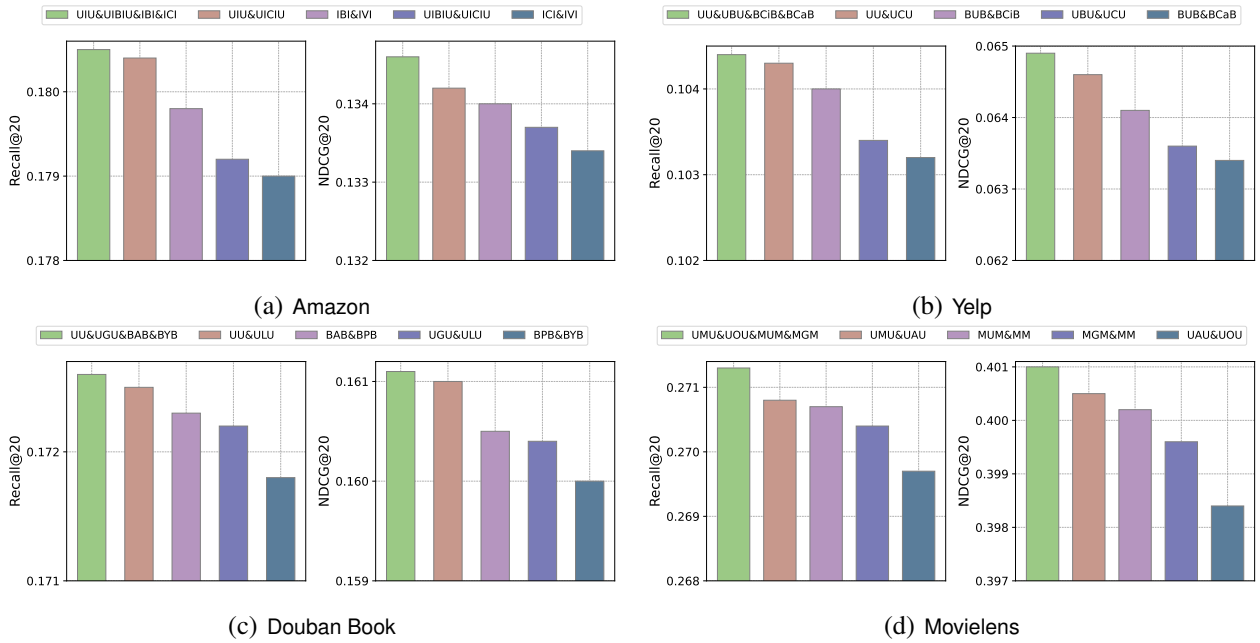



Fig. 9: The impact of different meta-path combinations. The green bar represents the optimal meta-path we used, while the others show the replaced meta-paths. **The minimum value on the $y$-axis is significantly better than strongest baselines.**

TABLE IV: Ablation studies of IHGCL on all datasets w.r.t. Recall@20 and NDCG@20.

| Datasets | Last.fm | | Amazon | | Yelp | | Douban Book | | Movielens | | Douban Movie | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metrics | R@20 | N@20 | R@20 | N@20 | R@20 | N@20 | R@20 | N@20 | R@20 | N@20 | R@20 | N@20 |
| w/o DCL | 0.2690 | 0.2657 | 0.1687 | 0.1237 | 0.0949 | 0.0580 | 0.1568 | 0.1451 | 0.2557 | 0.3882 | 0.1898 | 0.2124 |
| w/o BAE | 0.2754 | 0.2725 | 0.1742 | 0.1286 | 0.0952 | 0.0583 | 0.1589 | 0.1480 | 0.2623 | 0.3913 | 0.1950 | 0.2138 |
| w/o IB | 0.2787 | 0.2782 | 0.1781 | 0.1320 | 0.1010 | 0.0629 | 0.1714 | 0.1584 | 0.2687 | 0.4001 | 0.1966 | 0.2141 |
| w/o ICL | 0.2769 | 0.2766 | 0.1773 | 0.1313 | 0.1021 | 0.0632 | 0.1696 | 0.1593 | 0.2685 | 0.3998 | 0.1956 | 0.2121 |
| w/o IICL | 0.2677 | 0.2638 | 0.1713 | 0.1262 | 0.0961 | 0.0586 | 0.1543 | 0.1434 | 0.2604 | 0.3926 | 0.1921 | 0.2154 |
| IHGCL | **0.2824** | **0.2815** | **0.1795** | **0.1346** | **0.1044** | **0.0649** | **0.1726** | **0.1611** | **0.2713** | **0.4010** | **0.2003** | **0.2195** |

the independent intent-intent contrast weakens the model's capacity to capture user preferences via interactions.

Finally, the information bottleneck, as a crucial component of BAE, imposes constraints on the denoising objective of the autoencoder, thereby validating the effectiveness of the denoising strategy.

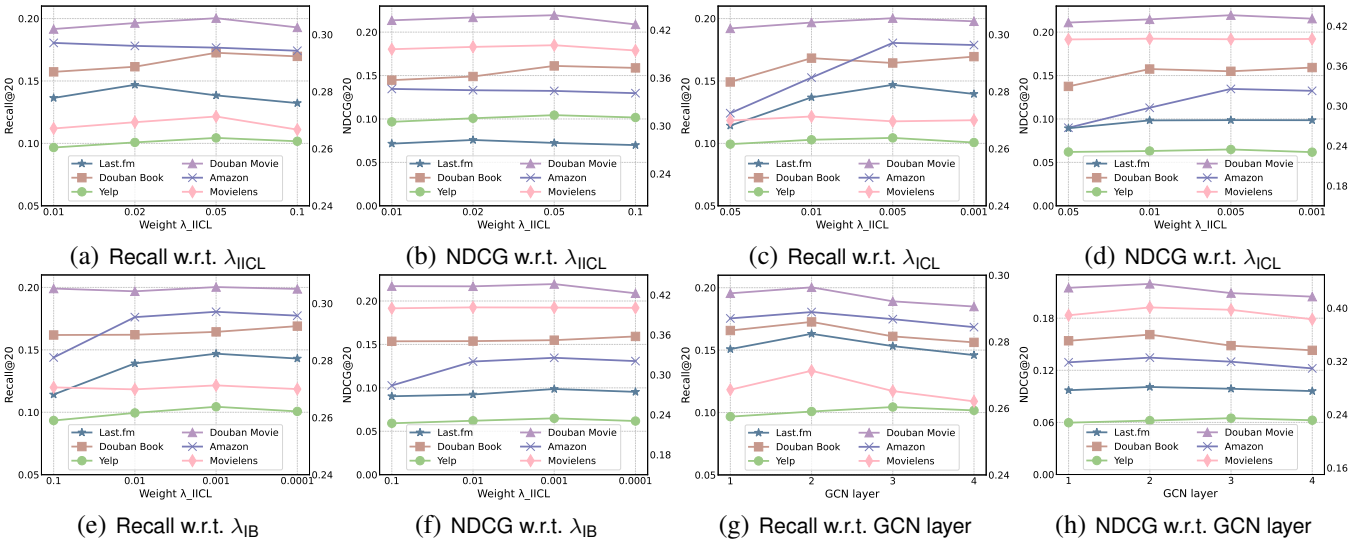
### *F. Hyperparameter Sensitivity (RQ5)*

In this study, we explored key hyperparameters' impact on our model. We investigated the effects of different losses, mask ratio in BAE and GCN layer for main task.

- **IICL coefficient.** Our ablation studies indicated that the IICL loss exerts the most significant impact on the experimental results. To investigate the model's sensitivity to the strength of control in contrastive learning using the loss, we analyzed its impact on model performance within the range $(0.01, 0.02, 0.05, 0.1)$ as illustrated in Figures 10 (a) and (b). It was observed that most datasets achieved optimal performance at 0.05 (or lower), and higher values of $\lambda_{\text{IICL}}$ tended to overly emphasize the contrastive optimization loss.
- **ICL coefficient.** As shown in Figures 10 (c) and (d), $\lambda_{\text{ICL}}$ achieves optimal performance at 0.005 for most datasets, except for Amazon at 0.001 and Movielens at 0.01. We attribute this phenomenon to the sparse nature of Amazon, which requires a lower degree of intent-intent alignment to allow intent-interaction to have a greater impact on the recommendation process.
- **IB coefficient.** As shown in Figures 10 (e) and (f), $\lambda_{\text{IB}}$ is more sensitive in smaller datasets such as Amazon and Last.fm, indicating that $\mathcal{L}_{\text{IB}}$'s effect needs to be diminished in scenarios with less noise. Conversely, $\mathcal{L}_{\text{IB}}$ exhibits stable contributions in larger datasets.
- **GCN layer.** The results from Figures 10 (c) and (d) suggest that when using two GCN layers in IHGCL, the model can effectively integrate the intents of users and items into the main task. However, stacking more GCN layers may cause noise issues, with this noise propagating to farther nodes through iterations, ultimately resulting in degradation.
- **Mask ratio.** In Section III-B, we employed a masked autoencoder with dual mask, where the mask ratio plays a crucial role in eliminating noise from heterogeneous information. The red points indicated in Figure 11 represent the optimal parameter values, and most datasets exhibit a prominent peak. We can observe that larger-scale datasets typically require a higher mask ratio for denoising, thus confirming the conclusion that semantic information contains a significant amount of noise.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3536096

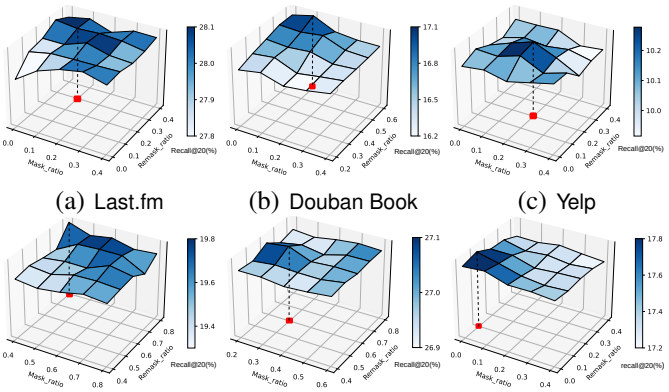





(a) Recall w.r.t. $\lambda_{\text{IICL}}$ (b) NDCG w.r.t. $\lambda_{\text{IICL}}$ (c) Recall w.r.t. $\lambda_{\text{ICL}}$ (d) NDCG w.r.t. $\lambda_{\text{ICL}}$

(e) Recall w.r.t. $\lambda_{\text{IB}}$ (f) NDCG w.r.t. $\lambda_{\text{IB}}$ (g) Recall w.r.t. GCN layer (h) NDCG w.r.t. GCN layer

Fig. 10: Performance comparison w.r.t. IICL coefficient and GCN layer. The axes on the right belong to the Last.fm and Movielens datasets and the left belong to other datasets.




(a) Last.fm (b) Douban Book (c) Yelp

(d) Douban Movie (e) Movielens (f) mazon

Fig. 11: Performance comparison of different mask ratio.

## VI. RELATED WORK

### A. Contrastive Learning for Recommendation

Contrastive learning [34], [42], [43] has emerged as one of the paradigms in self-supervised learning, which aims to learn representation invariants by pulling positive samples closer and pushing negative samples apart. Recently, some studies [9]–[11], [13], [44], [45] have explored the application of contrastive learning in recommender systems. These methods leverage various embedding contrasts to generate effective self-supervised signals to alleviate the data sparsity issue. In particular, SGL [10] designs three graph structure augmentation methods to construct contrastive views and maximizes the consistency between views to enhance recommendation. NCL [44] explicitly models the semantic information brought by user (or item) relationships and realizes contrast between nodes of the same type through the outputs of different layers of GNN. SimGCL [11] explores the effectiveness of graph structure augmentation and proposes a simple paradigm of noise-enhanced embedding for contrast. DirectAU [13] analyzes the alignment and uniformity properties of contrastive learning and effectively enhances the recommendation performance by using loss functions that optimize these two objectives. LightGCL [9] uses singular value decomposition (SVD) to construct lightweight contrastive views for recommendation.

### B. Heterogeneous Graph Neural Networks

Heterogeneous graphs (HGs) have unique advantages in modeling complex relationships in the real world. In recent years, Graph Neural Networks (GNNs) [46]–[48] have made significant progress in various fields, and inspired by this, Heterogeneous Graph Neural Networks (HGNNs) [19], [20], [30], [49]–[51] combine rich semantics indicated by different meta-paths with GNNs, achieving success in more complex graph structure domains. Specifically, these HGNNs can be classified into semi-supervised and self-supervised. Semi-supervised models include HAN [19], and MAGNN [50]. HAN employs a dual-layer attention mechanism at both the node and semantic levels to adaptively capture the relations. MAGNN integrates meta-paths and adaptive mechanisms to learn node representations on heterogeneous graphs by learning meta-path attention weights at both the node and global levels. The self-supervised models include HeCo [20], and HGMAE [51]. HeCo contrasts views across network schema and meta-path to achieve self-supervised augmentation. Meanwhile, HGMAE further introduces dynamic masking mechanisms and position-aware encoding to enhance the model's performance.

### C. Disentanglement-based Recommendation

Disentanglement-based methods generally focus on modeling user-item interactions by projecting them into distinct feature spaces [52], [53]. For instance, MacridVAE [54] leverages variational autoencoders to encode various user intents [55]. DGCF [16] employs graph neural networks to learn disentangled user representations. DisenHAN [17] utilizes meta-relation decomposition along with disentangled propagation layers to capture semantic meanings. In the case of CDR [56], a dynamic routing mechanism is developed to characterize the correlations among user intents for embedding denoising. KGIN [57] introduces the concept of shared intents and uses

an item-side knowledge graph to capture user's path-based intents. Some innovative approaches have started integrating contrastive learning into intent modeling, such as ICLRec [58], DCCF [12], and BIGCF [27]. DCCF [12] enhances self-supervised signals by learning disentangled representations with a global context, while BIGCF investigates the individuality and collectivity of intents behind interactions for collaborative filtering. Nevertheless, these approaches do not address the combination of fine-grained intents and contrastive learning within heterogeneous graphs. The proposed IHGCL seeks to replace unreliable data augmentation with modeled intent embeddings.

## VII. Conclusion

In this paper, we considered leveraging fine-grained intents of users and items in heterogeneous graph recommendation to construct augmented views for contrastive learning, and we represented these intents from an explainable standpoint using various meta-paths. Based on this, we proposed a novel end-to-end recommendation model: Intent-guided Heterogeneous Graph Contrastive Learning (IHGCL). We introduced dual contrastive learning to incorporate the captured intents behind meta-paths into the main user-item view, forming augmented contrastive views and aligning intents across meta-paths. Additionally, to alleviate the noise issues inherent in intents, we further proposed a bottlenecked autoencoder that combines mask and information bottleneck. Finally, we conducted extensive experiments on six real-world datasets to validate the effectiveness of IHGCL.

## Acknowledgment

This work is supported by the National Science Foundation of China (No. 62206002, No. 62272001 and No. 62206004), Hefei Key Common Technology Project (NO. 2023SGJ014), Natural Science Foundation of Anhui Province (No. 2208085QF195) and Xunfei Zhiyuan Digital Transformation Innovation Research Special for Universities (NO. 2023ZY001).

## References

[1] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Springer, 2010, pp. 1–35.

[2] J. Wu, J. Chen, J. Wu, W. Shi, J. Zhang, and X. Wang, "Bsl: Understanding and improving softmax loss for recommendation," in *ICDE*, 2024.

[3] L. Sang, M. Xu, S. Qian, M. Martin, P. Li, and X. Wu, "Context-dependent propagating-based video recommendation in multimodal heterogeneous information networks," *IEEE Transactions on Multimedia (TMM)*, vol. 23, pp. 2019–2032, 2020.

[4] C. Wu, F. Wu, Y. Huang, and X. Xie, "Personalized news recommendation: Methods and challenges," *ACM Transactions on Information Systems (TOIS)*, vol. 41, no. 1, pp. 1–50, 2023.

[5] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *SIGIR*, 2020, pp. 639–648.

[6] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[7] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *SIGIR*, 2019, pp. 165–174.

[8] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW*, 2017, pp. 173–182.

[9] X. Cai, C. Huang, L. Xia, and X. Ren, "Lightgcl: Simple yet effective graph contrastive learning for recommendation," in *ICLR*, 2023.

[10] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *SIGIR*, 2021, pp. 726–735.

[11] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, and Q. V. H. Nguyen, "Are graph augmentations necessary? simple graph contrastive learning for recommendation," in *SIGIR*, 2022, pp. 1294–1303.

[12] X. Ren, L. Xia, J. Zhao, D. Yin, and C. Huang, "Disentangled contrastive collaborative filtering," in *SIGIR*, 2023, pp. 1137–1146.

[13] C. Wang, Y. Yu, W. Ma, M. Zhang, C. Chen, Y. Liu, and S. Ma, "Towards representation alignment and uniformity in collaborative filtering," in *KDD*, 2022, pp. 1816–1825.

[14] Y. Sun and J. Han, "Mining heterogeneous information networks: A structural analysis approach," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 20–28, 2013.

[15] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, and R. Luo, "Heterogeneous graph contrastive learning for recommendation," in *WSDM*, 2023, pp. 544–552.

[16] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, and T.-S. Chua, "Disentangled graph collaborative filtering," in *SIGIR*, 2020, pp. 1001–1010.

[17] Y. Wang, S. Tang, Y. Lei, W. Song, S. Wang, and M. Zhang, "Disenhan: Disentangled heterogeneous graph attention nnetwork for recommendation," in *CIKM*, 2020, pp. 1605–1614.

[18] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *KDD*, 2017, pp. 135–144.

[19] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *WWW*, 2019, pp. 2022–2032.

[20] X. Wang, N. Liu, H. Han, and C. Shi, "Self-supervised heterogeneous graph neural network with co-contrastive learning," in *KDD*, 2021, pp. 1726–1736.

[21] D. Cai, S. Qian, Q. Fang, J. Hu, W. Ding, and C. Xu, "Heterogeneous graph contrastive learning network for personalized micro-video recommendation," *IEEE Transactions on Multimedia (TMM)*, vol. 25, pp. 2761–2773, 2023.

[22] Q. Zhang, L. Xia, X. Cai, S. Yiu, C. Huang, and C. S. Jensen, "Graph augmentation for recommendation," in *ICDE*, 2024.

[23] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip, "Heterogeneous information network embedding for recommendation," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 31, no. 2, pp. 357–370, 2018.

[24] H. Wang, K. Zhou, X. Zhao, J. Wang, and J.-R. Wen, "Curriculum pre-training heterogeneous subgraph transformer for top-n recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 41, no. 1, pp. 1–28, 2023.

[25] C. Yang, X. Gong, C. Shi, and P. Yu, "A post-training framework for improving heterogeneous graph neural networks," in *WWW*, 2023, pp. 251–262.

[26] J. Zheng, Q. Ma, H. Gu, and Z. Zheng, "Multi-view denoising graph auto-encoders on heterogeneous information networks for cold-start recommendation," in *KDD*, 2021, pp. 2338–2348.

[27] Y. Zhang, L. Sang, and Y. Zhang, "Exploring the individuality and collectivity of intents behind interactions for graph collaborative filtering," in *SIGIR*, 2024, pp. 1253–1262.

[28] Y. Zhang, Y. Zhang, D. Yan, S. Deng, and Y. Yang, "Revisiting graph-based recommender systems from the perspective of variational auto-encoder," *ACM Transactions on Information Systems (TOIS)*, vol. 41, no. 3, pp. 1–28, 2023.

[29] X. Ren, W. Wei, L. Xia, and C. Huang, "A comprehensive survey on self-supervised learning for recommendation," *arXiv preprint arXiv:2404.03354*, 2024.

[30] Z. Wang, Q. Li, D. Yu, X. Han, X.-Z. Gao, and S. Shen, "Heterogeneous graph contrastive multi-view learning," in *SDM*. SIAM, 2023, pp. 136–144.

[31] Q. Sun, J. Li, H. Peng, J. Wu, X. Fu, C. Ji, and S. Y. Philip, "Graph structure learning with variational information bottleneck," in *AAAI*, vol. 36, no. 4, 2022, pp. 4165–4174.

[32] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[33] C. Wei, J. Liang, D. Liu, and F. Wang, "Contrastive graph structure learning via information bottleneck for recommendation," *NeurIPS*, vol. 35, pp. 20 407–20 420, 2022.

[34] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*. PMLR, 2020, pp. 1597–1607.

[35] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[36] X. Long, C. Huang, Y. Xu, H. Xu, P. Dai, L. Xia, and L. Bo, "Social recommendation with self-supervised metagraph informax network," in *CIKM*, 2021, pp. 1160–1169.

[37] Y. Zhang, Y. Zhang, Y. Zhao, S. Deng, and Y. Yang, "Dual variational graph reconstruction learning for social recommendation," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2024.

[38] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics (ICAIS)*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[39] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *ICML*. PMLR, 2020, pp. 9929–9939.

[40] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *JMLR*, vol. 9, no. 11, 2008.

[41] Y. Ma, Y. He, A. Zhang, X. Wang, and T.-S. Chua, "Crosscbr: Cross-view contrastive learning for bundle recommendation," in *KDD*, 2022, pp. 1233–1241.

[42] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020, pp. 9729–9738.

[43] L. Chen, L. Wu, K. Zhang, R. Hong, D. Lian, Z. Zhang, J. Zhou, and M. Wang, "Improving recommendation fairness via data augmentation," in *WWW*, 2023, p. 1012–1020.

[44] Z. Lin, C. Tian, Y. Hou, and W. X. Zhao, "Improving graph collaborative filtering with neighborhood-enriched contrastive learning," in *WWW*, 2022, pp. 2320–2329.

[45] J. Yu, X. Xia, T. Chen, L. Cui, N. Q. V. Hung, and H. Yin, "Xsimgcl: Towards extremely simple graph contrastive learning for recommendation," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 36, no. 2, pp. 913–926, 2023.

[46] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, and J. Tang, "Graphmae: Self-supervised masked graph autoencoders," in *KDD*, 2022, pp. 594–604.

[47] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun, "Gpt-gnn: Generative pre-training of graph neural networks," in *KDD*, 2020, pp. 1857–1867.

[48] L. Yu, L. Sun, B. Du, C. Liu, W. Lv, and H. Xiong, "Heterogeneous graph representation learning with relation awareness," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 35, no. 6, pp. 5935–5947, 2022.

[49] L. Sang, Y. Wang, Y. Zhang, and X. Wu, "Denoising heterogeneous graph pre-training framework for recommendation," *ACM Transactions on Information Systems (TOIS)*, 2024.

[50] X. Fu, J. Zhang, Z. Meng, and I. King, "Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding," in *WWW*, 2020, pp. 2331–2341.

[51] Y. Tian, K. Dong, C. Zhang, C. Zhang, and N. V. Chawla, "Heterogeneous graph masked autoencoders," in *AAAI*, vol. 37, no. 8, 2023, pp. 9997–10 005.

[52] S. Fan, J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, and Y. Li, "Metapath-guided heterogeneous graph neural network for intent recommendation," in *KDD*, 2019, pp. 2478–2486.

[53] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *WWW*, 2018, pp. 689–698.

[54] J. Ma, C. Zhou, P. Cui, H. Yang, and W. Zhu, "Learning disentangled representations for recommendation," *NeurIPS*, vol. 32, 2019.

[55] J. Ma, P. Cui, K. Kuang, X. Wang, and W. Zhu, "Disentangled graph convolutional networks," in *ICML*. PMLR, 2019, pp. 4212–4221.

[56] H. Chen, Y. Chen, X. Wang, R. Xie, R. Wang, F. Xia, and W. Zhu, "Curriculum disentangled recommendation with noisy multi-feedback," *NeurIPS*, vol. 34, pp. 26 924–26 936, 2021.

[57] X. Wang, T. Huang, D. Wang, Y. Yuan, Z. Liu, X. He, and T.-S. Chua, "Learning intents behind interactions with knowledge graph for recommendation," in *WWW*, 2021, pp. 878–887.

[58] Y. Chen, Z. Liu, J. Li, J. McAuley, and C. Xiong, "Intent contrastive learning for sequential recommendation," in *WWW*, 2022, pp. 2172–2182.

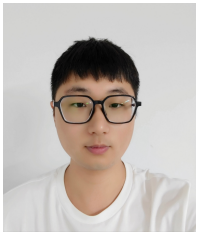
[59] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *KDD*, 2019, pp. 793–803.

[60] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *NeurIPS*, vol. 33, pp. 21 271–21 284, 2020.

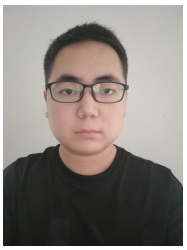
[61] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 35, no. 1, pp. 857–876, 2021.

[62] D. Zhang, Y. Geng, W. Gong, Z. Qi, Z. Chen, X. Tang, Y. Shan, Y. Dong, and J. Tang, "Recdcl: Dual contrastive learning for recommendation," in *WWW*, 2024, p. 3655–3666.


**Lei Sang** received the Ph.D. degree from the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, Australia, in 2021. He is currently a Lecturer with the School of Computer Science and Technology, Anhui University, Anhui, China. His current research interests include natural language processing, data mining, and recommender systems.



**Yu Wang** received the Bachelor degree in Computer Science and Technology from Fuyang Normal University, Anhui, China, in 2022. He is currently pursuing the master degree at Anhui University's School of Computer Science and Technology. His current research interests include graph neural network, recommender systems, and data mining.



**Yi Zhang** received the Bachelor degree and the Master degree in Computer Science and Technology from Anhui University, Hefei, China, in 2020 and 2023, respectively. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Anhui University, Hefei, China. He has more than ten publications in several flagship journals and conferences, including IEEE TKDE, IEEE TSMC, IEEE TBD, ACM TOIS, and ACM SIGIR, etc. His current research interests include graph neural network, personalized recommender systems, and service computing.

**Yiwen Zhang** received the Ph.D. degree in management science and engineering from Hefei University of Technology, in 2013. He is currently a full professor with the School of Computer Science and Technology, Anhui University. He has published more than 100 papers in highly regarded conferences and journals, including IEEE TKDE, IEEE TMC, IEEE TSC, ACM TOIS, IEEE TPDS, IEEE TNNLS, ACM TKDD, SIGIR, ACL etc. His research interests include service computing, recommender systems,and big data analytics.

**Xindong Wu** (Fellow, IEEE) received the B.S. and M.S. degrees in computer science from the Hefei University of Technology, in 1987, and the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K., in 1993. He is currently the Director and Professor with the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, Hefei, China, and a Senior Research Scientist with the Research Center for Knowledge Engineering, Zhejiang Lab, China. He is a Foreign Member of Russian Academy of Engineering, and a Fellow of AAAS. His research interests include data mining, knowledge engineering, Big Data analytics, and marketing intelligence.