Heterogeneous Graph Contrastive Learning With Metapath-Based Augmentations

Xiaoru Chen, Yingxu Wang, Jinyuan Fang, Zaiqiao Meng, and Shangsong Liang

***Abstract*—Heterogeneous graph contrastive learning is an effective method to learn discriminative representations of nodes in heterogeneous graph when the labels are absent. To utilize metapath in contrastive learning process, previous methods always construct multiple metapath-based graphs from the original graph with metapaths, then perform data augmentation and contrastive learning on each graph respectively. However, this paradigm suffers from three defects: 1) It does not consider the augmentation scheme on the whole metapath-based graph set, which hinders them from fully leveraging the information of metapath-based graphs to achieve better performance. 2) The final node embeddings are not optimized from the contrastive objective directly, so they are not guaranteed to be distinctive enough. It leads to suboptimal performance on downstream tasks. 3) Its computational complexity for contrastive objective is high. To tackle these defects, we propose a Heterogeneous Graph Contrastive learning model with Metapath-based Augmentations (HGCMA), which is designed for downstream tasks with a small amount of labeled data. To address the first defect, both semantic-level and node-level augmentation schemes are proposed in our HGCMA for augmentation, where a metapath-based graph and a certain ratio of edges in each metapath-based graph are randomly masked, respectively. To address the second and third defects, we utilize a two-stage attention aggregation graph encoder to output final node embedding and optimize them with contrastive objective directly. Extensive experiments on three public datasets validate the effectiveness of HGCMA when compared with state-of-the-art methods.**

***Index Terms*—Graph neural network, graph representation learning, contrastive learning.**

I. Introduction

In a wide spectrum of applications in data mining, data can be intuitively cast into heterogeneous graphs, such as bibliographic information network [1], where entities are represented by different types of nodes or/and the relations among the entities are represented by different types of edges. Graph Neural Networks (GNNs) [2], [3] in particular are effective techniques to learn node representations by aggregating the representations between each node and its neighbors at each layer. Modeling heterogeneous graphs with GNNs has been proven to be effective in many applications such as node classification [4], [5], [6] and recommendation systems [7], [8], [9]. However, most GNNs require huge label data for training, which in many cases is difficult to obtain. To alleviate the labeling workload, graph contrastive learning has been proposed [10], [11]. It learns node representations by maximizing the similarities among positive samples while minimizing the similarities among negative ones. Both positive and negative samples are extracted from the unlabeled data. Therefore, graph contrastive learning is an effective self-supervised learning method to learn discriminative embeddings when the labels are absent. Considering the scarcity of labeled data, in this article, we leverage contrastive learning to tackle the representation learning problem in heterogeneous graphs for downstream tasks with a small amount of labeled data.

To effectively infer representations of nodes in heterogeneous graphs, some contrastive heterogeneous graph representation learning methods [12], [13], [14], [15], [16] have been proposed. To fully exploit the semantic information in heterogeneous graphs, they often utilize metapath to extend the original graph data. Metapath [17] is a sequence of relations connecting two node types (e.g., Fig. 1(c)), and is widely used to represent the semantic information in heterogeneous graphs (The definition of metapath is described in details in Section III). Specifically, in a heterogeneous graph, each metapath can be viewed as a type of high-order relation. A metapath-based graph can be constructed based on this relation, purely representing this high-order relation in the heterogeneous graph (The definition of metapath-based graph is described in details in Section III). Based on metapaths, most of them follow a multi-graph paradigm: They construct multiple metapath-based graphs from the original graph with metapaths. Then, in each training epoch, they perform data augmentation and contrastive learning on each graph respectively. The final embeddings are obtained by applying mean pooling or attention mechanism over node embeddings from each metapath-based graph. However, this paradigm suffers from the following defects: (1) It only performs data augmentation on each metapath-based graph separately and ignores considering the augmentation scheme

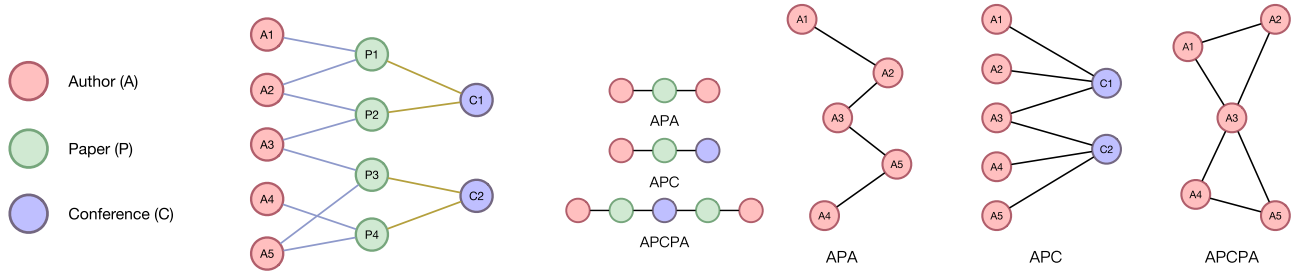
This work was supported in part by the National Natural Science Foundation of China under Grant 61906219 and in part by MBZUAI. *(Corresponding author: Shangsong Liang.)*

Xiaoru Chen is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mail: chenxr27@mail2.sysu.edu.cn).

Yingxu Wang is with the Department of Machine Learning, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi 7909, UAE (e-mail: yingxv.wang@gmail.com).

Jinyuan Fang and Zaiqiao Meng are with the School of Computing Science, University of Glasgow, G12 8QQ Glasgow, U.K. (e-mail: fangjy6@gmail.com; zaiqiao.meng@gmail.com).

Shangsong Liang is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China, and also with the Department of Machine Learning, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi 7909, UAE (e-mail: liangshangsong@gmail.com).



(a) Node Types (b) Heterogeneous Graph (c) Metapath (d) Metapath-based Graph

Fig. 1. Illustrative example of a heterogenous graph (a bibliographic information network). (a) Three types of nodes (i.e., Author, Paper, and Conference) in this graph. (b) Structure of the heterogenous graph. (c) Three metapaths involved in the graph (i.e., *APA*, *APC* and *APCPA*). (d) The metapath-based graphs constructed with the graph and metapaths.

on the whole metapath-based graph set, which hinders it from fully leveraging the information of metapath-based graphs to achieve better performance. (2) It mainly employs contrastive objectives on each metapath-based graph separately. Thus, the final node embeddings may lack sufficient distinctiveness since they are not optimized from the contrastive objectives directly. This leads to suboptimal performance on downstream tasks. (3) In contrastive optimization, it needs to compute the similarity of each node pair in each metapath-based graph, which leads to a heavy computational burden.

To tackle the aforementioned defects, we propose a **H**eterogeneous **G**raph **C**ontrastive learning model with **M**etapath-based **A**ugmentations, named HGCMA.[1] Following previous work, to exploit the semantic information in heterogeneous graphs, we utilize metapaths to construct corresponding metapath-based graphs. These metapath-based graphs as a whole serve as the extended view of the original graph. Based on the extended view, we propose two augmentation schemes to construct augmented views. The first one is a semantic-level augmentation method, namely Metapath-based Graph Mask, where we randomly mask a graph among the metapath-based graphs. The second one is a node-level augmentation method, namely Metapath-based Neighbor Mask, where we randomly mask a certain ratio of edges in each metapath-based graph. To obtain node embedding from the extended view, a two-stage aggregation graph encoder is leveraged. For each metapath-based graph, we adopt a graph attention layer to obtain its node embedding. Then we adopt graph-level attention to automatically learn the importance of different metapath-based graphs and fuse the corresponding node embeddings to get final embeddings. In each training epoch, we generate two augmented views by performing augmentations on the extended view, and obtain node embeddings in these two augmented views with the same graph encoder respectively. Although the mechanism for the generation of the two sets of node embeddings is the same, the values of embeddings are not the same due to the randomness of the data augmentations. Then we train the model using a contrastive objective to maximize the agreement between the final node embeddings in these two views directly.

[1]The code resource of our proposed HGCMA model is available for download at https://github.com/Chenxr1997/HGCMA.

In our method, the two data augmentation schemes are based on the metapaths of heterogeneous graph, which deeply involves the inherent information of heterogeneous graph in the contrastive training process. And with Metapath-based Graph Mask, we leverage the information of the whole extended view to achieve better performance. And our final node embeddings are optimized from the contrastive objective directly, which improves the distinctiveness of node embeddings and reduces the computational burden.

In summary, our contributions include:

- We propose a novel metapath-based graph contrastive learning method for heterogeneous graph representation learning task. The proposed method extends the original graph data by constructing metapath-based graphs. Based on the extended view, we propose two graph data augmentation schemes: Metapath-based Graph Mask and Metapath-based Neighbor Mask, which leverage the information of the whole extended view to further improve the performance.
- We propose to perform contrastive optimization directly on final node embeddings, enhancing the distinctiveness of node embeddings and reducing the computational complexity.
- We conduct extensive experiments on three public datasets to validate the effectiveness of HGCMA and the correctness of our motivation. Our ablation study further demonstrates that our augmentation schemes are beneficial for heterogeneous graph contrastive learning.

II. Related Work

Here, we briefly describe two lines of related work, heterogeneous graph representation learning and graph contrastive learning.

A. Heterogeneous Graph Representation Learning

Heterogeneous graph representation learning aims to obtain meaningful node representations in heterogeneous graphs to facilitate various downstream tasks [18], such as node classification and link prediction [12], [19]. Most of the existing methods for this task can be generally classified into two categories: 1) proximity-preserving methods [4], [20], [21];

2) message-passing methods [5], [22], [23]. Proximity-preserving methods directly encode each node in a heterogeneous graph as a vector and learn the vectors by making the positive node pairs closer to each other than the negative node pairs. There are two major methods to collect positive node pairs: metapath-guided random walk [4], [24], [25] and constructing first/second-order proximity-based node pairs [20], [26], [27]. Message-passing methods aim to learn node embeddings by aggregating the information from their neighbors, and graph neural networks (GNNs) [2], [3] are widely used for it. To adopt GNNs for heterogeneous graphs, edge heterogeneity [22] and metapath [5], [6], [23] are considered in node representation aggregation. HAE [28] simultaneously incorporates metapaths and metagraphs, and leverages the self-attention mechanism to explore content-based nodes' interactions. HPN [29] improves the node-level aggregating process via absorbing node's local semantic with a proper weight.

B. Graph Contrastive Learning

Contrastive Learning (CL) [30], [31], [32] was first proposed in computer vision to train CNNs for image representation learning. Its key idea is to contrast semantically similar (positive) and dissimilar (negative) pairs of images, aiming to maximize the mutual information (MI) between positive pairs.

Graph Contrastive Learning (GCL) applies the idea of CL to GNNs, to learn representations on graphs without annotations. As summarized in [33], GCL constructes multiple views of a graph and formulates contrastive learning tasks on these views with the help of a contrastive objective function. As a pioneering work, DGI [34] constructs negative graph views by random shuffling node attributes and utilizes InfoMAX [35] to maximize mutual information between node embeddings and global summary embeddings. Following the framework of DGI, MVGRL [36] proposes to generate graph diffusions as graph views and GraphCL [10] proposes various basic augmentation methods to construct graph views. Xu et al. [37] propose a group contrastive learning framework to contrast multiple representations in various subspaces. Inspired by instance discrimination [38], GRACE [39] and GCA [40] eschew the need of graph embeddings and propose node-level contrastive tasks. Zhang et al. [41] further propose to perform augmentations on the hidden features (feature augmentation) to reduce the bias of node embeddings.

Additionally, several approaches have been proposed for Heterogeneous Graph Contrastive Learning (HGCL). Methods in this domain typically utilize metapaths to construct metapath-based graphs and perform contrastive learning on them. DMGI [13] extends DGI to each metapath-based graph and train consensus vectors as node representations. Wang et al. [12] employ network schema and metapath views to collaboratively supervise each other. HORACE [15] maximizes the mutual information between each metapath-based graph view and aggregated view, and HGCML [16] further maximizes the mutual information between any pairs of metapath-based graph views. X-GOAL [14] not only performs contrastive learning on each metapath-based graph, but also introduces cluster-level alignment to pull nodes with similar semantic closer. Besides constructing metapath-based graphs, CPT-HG [42] propose to leverage both relation- and metagraph-level neighbors to collect positive samples. However, these methods do not fully utilize the information contained within metapath-based graphs. For graph view construction, they either directly construct metapath-based graphs or perform data augmentation on them separately, neglecting the augmentation scheme on the entire set of metapath-based graphs, which hinders them from fully leveraging the information of metapath-based graphs. For contrastive optimization, most of them employ contrastive objective on each metapath-based graph separately, which leads to a heavy computational burden and suboptimal performance on downstream tasks.

III. Preliminaries

In this section, we provide the definitions used across this article and the task to be addressed.

A. Definitions of Important Concepts

Many real-world data are interconnected and therefore can be represented as heterogeneous graphs, i.e., graphs with various types of nodes and edges, such as bibliographic information networks. Formally, the heterogeneous graphs can be defined as:

Definition 1. Heterogeneous Graph: A heterogeneous graph is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of nodes and edges, respectively. The number of nodes and edges are represented as $|\mathcal{V}|$ and $|\mathcal{E}|$, respectively. Nodes and edges are associated with a node type mapping function $\phi : \mathcal{V} \to \mathcal{T}$ and an edge type mapping function $\psi : \mathcal{E} \to \mathcal{R}$ respectively, where $\mathcal{T}$ and $\mathcal{R}$ denote the sets of node and edge types respectively, and $|\mathcal{T}| > 1$ and/or $|\mathcal{R}| > 1$ (there are more than one type of nodes or edges in $\mathcal{G}$).

Example: Fig. 1(b) shows an example of heterogeneous graphs (bibliographic information network) that comprises three node types: Author, Paper and Conference.

Definition 2. Metapath: A metapath Φ is defined as a pattern of paths in the form of $T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} T_{l+1}$ (abbreviated as $T_1T_2 \ldots T_{l+1}$), where $T_i \in \mathcal{T}, R_j \in \mathcal{R}$. The metapath starts at T_1 and describes a composite relation $R = R_1 \circ R_2 \circ \ldots \circ R_l$ between node type T_1 and T_{l+1}, where $\circ$ denotes the composition operator on relations.

Example: Fig. 1(c) shows three types of metapath starting at $Author$ in Fig. 1(b). The metapath APA is a path pattern, which links an author node to another author node through a paper node. Therefore, a path (e.g. $A_2 - P_1 - A_1$) following this pattern denotes that the two authors (i.e. A_1 and A_2) have co-author relationship. Similarly, a path (e.g. $A_3 - P_3 - C_2 - P_4 - A_5$) following APCPA denotes that the two authors (i.e., A_3 and A_5) have published papers in the same conference.

Definition 3. Metapath-based Relation: Given a metapath Φ and a heterogeneous graph $\mathcal{G}$, two nodes have a metapath-Φ-based relation with each other if there exists any path connecting

them and following the pattern defined by Φ, and they constitute a metapath-Φ-based relation pair.

Example: In Fig. 1(b), starting from A_2, there are two paths matching the pattern of metapath *APA*: $A_2 - P_1 - A_1$ and $A_2 - P_2 - A_3$. Thus, (A_2, A_1) and (A_2, A_3) are metapath-*APA*-based relation pairs. Starting from A_5, we can find a path matching the pattern of metapath *APCPA*: $A_5 - P_4 - C_2 - P_3 - A_3$, so (A_5, A_3) is a metapath-*APCPA*-based relation pair.

Definition 4. Metapath-based Graph: Given a metapath Φ and a heterogeneous graph $\mathcal{G}$, the metapath-based graph $\mathcal{G}^{\Phi}$ is a graph constructed by all the metapath-Φ-based relation pairs in graph $\mathcal{G}$.

Example: As shown in Fig. 1(d), we illustrate the metapath-based graph for three metapaths (*APA*, *APC* and *APCPA*), they are constructed by all of their respective metapath-based relation pairs.

Definition 5. Metapath-based Neighbor: Given a metapath Φ and a heterogeneous graph $\mathcal{G}$, $\mathcal{N}_i^{\Phi}$ is the set of all the neighbors of node i in metapath-Φ-based graph $\mathcal{G}^{\Phi}$.

Example: As shown in Fig. 1(d), the metapath-based neighbors of A_2 in *APA* are $\{A_1, A_3\}$ and the metapath-based neighbors of A_3 in *APCPA* are $\{A_1, A_2, A_4, A_5\}$.

B. The Task

The heterogeneous graph representation learning task can be formally defined as: given a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as defined in Definition 1, infer embeddings of the nodes. Specifically, we aim at seeking the function F_{X} that satisfies the following:

$$\mathcal{G} \xrightarrow{F_{\mathrm{X}}} \mathbf{Z}, \tag{1}$$

where $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{|\mathcal{V}|}] \in \mathbb{R}^{|\mathcal{V}| \times d}$ is a embedding matrix of the nodes and d is the dimension of the node embeddings ($d \ll |\mathcal{V}|$). The graph property is preserved as much as possible in $\mathbf{Z}$.

IV. Methodology

In this section, we detail our heterogeneous graph contrastive learning method, HGCMA. In particular, in Section IV-A, we provide an overview of our contrastive learning method; in Section IV-B, we propose a method to extend the original graph, after which an extended view of the graph is obtained; in Section IV-C, we propose our graph augmentation schemes on the above extended view; in Section IV-D, we obtain our node embeddings from the augmented views with the two-stage aggregation; in Section IV-E we learn node embeddings with graph contrastive learning; in Section IV-F we analysis the computational complexity of HGCMA.

A. Overview

An overview of our proposed method is provided in Fig. 2. Before training, to effectively exploit the semantic information in the graphs, we utilize metapaths to extend the original heterogeneous graph and get the extended view. At each training iteration, we first generate two augmented views by performing augmentations on the extended view, and then we utilize a graph encoder to output node embeddings for each augmented view. After the encoding process, with the help of contrastive objective, we optimize our model by distinguishing the same nodes from the others. The method of generating the node embeddings for one augmented view is illustrated in Fig. 3. To extend the original heterogeneous graph, we construct multiple metapath-based graphs (the subfigure (a2) of Fig. 3; detailed in Section IV-B), which are regarded as the extended view of the original heterogeneous graphs. Then, to generate the augmented view, we propose two novel semantic-level and node-level augmentation methods, where we randomly mask a metapath-based graph and a proportion of edges in each remaining metapath-based graph (the subfigures (b1) and (b2) of Fig. 3; detailed in Section IV-C). Next, we employ a two-stage aggregation graph encoder to produce node embeddings. The encoder first uses graph attention layers to generate node embeddings for each metapath-based graph. Subsequently, for each node, it aggregates its embedding across the metapath-based graphs with attention mechanism (subfigures (c1) and (c2) of Fig. 3; detailed in Section IV-D).

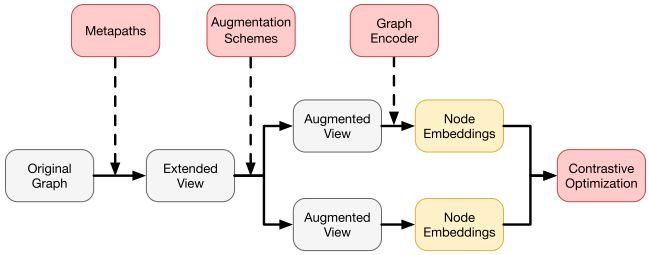


Fig. 2. Overview of HGCMA.

Following the previous work [12], in what follows we only discuss the training process of nodes with type A, and the representation of other nodes can be learned with the same training process.

B. Construction of Extended View

To fully extract structural and semantic information of a heterogeneous graph $\mathcal{G}$, we propose to extend the original graph with metapaths. Specifically, given a set of metapaths $\mathcal{P} = \{\Phi_1, \ldots, \Phi_P\}$, we construct the metapath-based graph for each metapath. We can regard the set of all these metapath-based graphs as the extended view of $\mathcal{G}$, denoted as $\mathcal{S} = \{\mathcal{G}^{\Phi_1}, \ldots, \mathcal{G}^{\Phi_P}\}$.

To learn the embeddings of nodes of type A, we utilize the subset of $\mathcal{P}$ in which the metapaths start at A, denoted as $\mathcal{P}_A = \{\Phi_1, \ldots, \Phi_{P_A}\}$, and the corresponding extended view is $\mathcal{S}_A$.

C. Augmentations Schemes

In graph contrastive learning, different graph views are required for the contrastive task, which is predicting the relationship among nodes from different views. With the extended graph view, we propose two augmentation schemes executed successively on $\mathcal{S}_A$ to produce augmented views for contrastive learning.

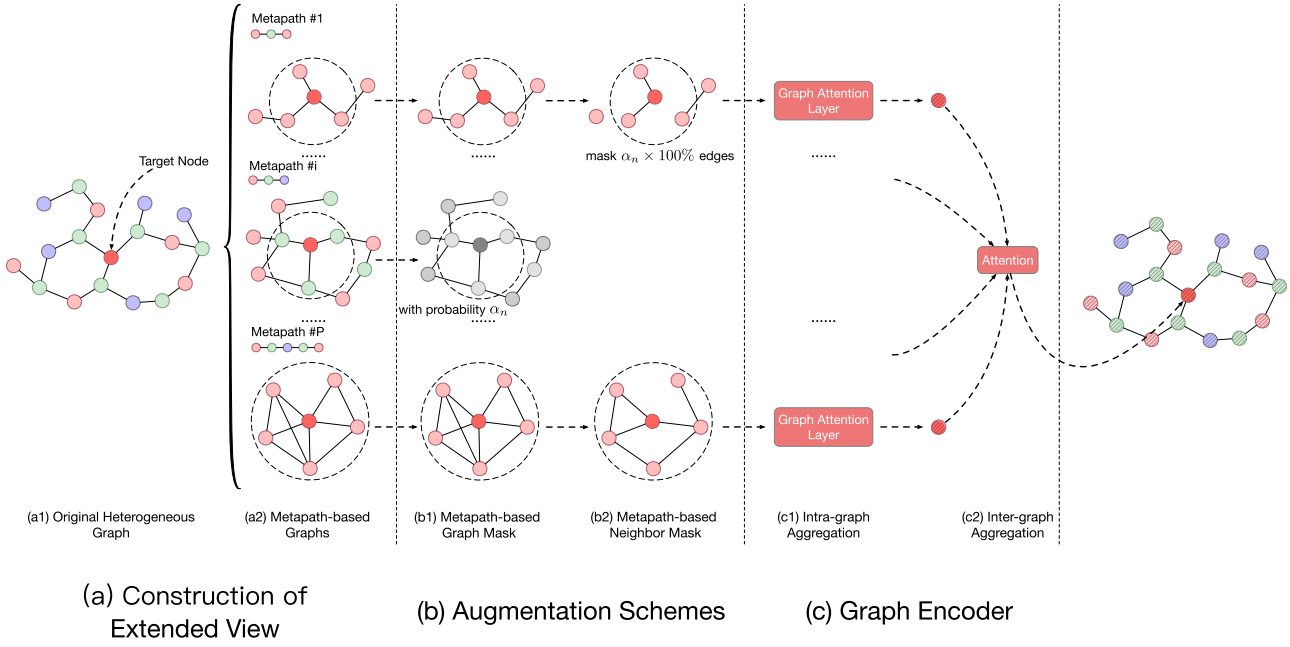


Fig. 3. Method of generating node embeddings at training stage in HGCMA. HGCMA takes a heterogeneous graph as input and constructs several metapath-based graphs as the extended view. To obtain augmented views, HGCMA performs data augmentation on the extended view by masking one metapath-based graph with probability α_g and masking $\alpha_n \times 100\%$ edges in each metapath-based graph. Then, HGCMA obtains node embeddings in augmented view by applying graph attention layer on each metapath-based graph and aggregating node embeddings from all the metapath-based graphs.

Algorithm 1: Augmentation Process.

Input: extended graph view $\mathcal{S}_A = \{\mathcal{G}^{\Phi_1}, \mathcal{G}^{\Phi_2}, \dots \mathcal{G}^{\Phi_{P_A}}\}$, the metapath-based graph masking probability α_g, the metapath-based neighbor masking ratio α_n
Output: augmented view $\widetilde{\mathcal{S}_A}$

Sample r from $\text{Bernoulli}(\alpha_g)$
if $r = 1$ **then**
  Randomly remove a metapath-based graph $\mathcal{G}^{\Phi}$ from $\mathcal{S}_A$;
end if
$\widetilde{\mathcal{S}_A} \leftarrow \emptyset$;
for metapath-based graph $\mathcal{G}^{\Phi} \in \mathcal{S}_A$ **do**
  Construct $\widetilde{\mathcal{G}^{\Phi}}$ by masking $\alpha_n\%$ edges in $\mathcal{G}^{\Phi}$;
  Add $\widetilde{\mathcal{G}^{\Phi}}$ to $\widetilde{\mathcal{S}_A}$.
end for

1) Metapath-Based Graph Mask: In the first augmentation scheme, we mask (i.e., remove) metapath-based graph in $\mathcal{S}_A$. Considering it is a coarse-grained augmentation, we only execute it with a certain probability in each training epoch and only mask one metapath-based graph in each execution. By changing the probability, we can control the degree of perturbation in the whole training process to adapt different graphs.

To be specific, in each training epoch, we sample a number r from a Bernoulli distribution $\text{Bernoulli}(\alpha_g)$, where α_g is the masking probability. If r is 1, we randomly choose one metapath-based graph from $\mathcal{S}_A$ with uniform sampling, and remove it from $\mathcal{S}_A$; otherwise, we do not perform this augmentation.

2) Metapath-Based Neighbor Mask: For metapath-based neighbor mask, we randomly mask (i.e., remove) edges in metapath-based graphs. Specifically, for each metapath-based graph $\mathcal{G}^{\Phi}$ in $\mathcal{S}_A$, we randomly mask (i.e., remove) $\alpha_n \times 100\%$ edges in $\mathcal{G}^{\Phi}$, where α_n is the masking ratio, and we denote the corrupted metapath-based graph as $\widetilde{\mathcal{G}^{\Phi}}$.

In each training epoch, when constructing an augmented view, we first perform metapath-based graph masking on the extended view, and then we perform metapath-based neighbor masking on the intermediate result, the process of which is shown in Algorithm 1. And we denote the corrupted graph view as $\widetilde{\mathcal{S}_A}$. The size of $\widetilde{\mathcal{S}_A}$ is either $|\mathcal{S}_A|$ or $|\mathcal{S}_A| - 1$, depending on whether the metapath-based graph mask is performed, and the elements in $\widetilde{\mathcal{S}_A}$ are the corrupted metapath-based graphs. By masking metapath-based graph and metapath-based neighbors, we utilize the information of the extended view in our augmentation schemes. Particularly, we leverage the information of the whole extended view with Metapath-based Graph Mask.

D. Graph Encoder

After the augmented views of a heterogeneous graph are obtained, we further introduce a graph encoder to obtain the final node embeddings, which consists of three components: node feature transformation, intra-graph aggregation, and inter-graph aggregation.

1) Node Feature Transformation: Since there are different types of nodes in a heterogeneous graph, features of nodes may lie in different feature spaces. Therefore, to unify different feature spaces, for an arbitrary node $i \in \mathcal{V}_{A'}$ of type $A' \in \mathcal{A}$, we

have

$$\mathbf{h}_i = \sigma(\mathbf{W}_{A'} \cdot \mathbf{x}_i + \mathbf{b}_{A'}), \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^{d_{A'}}$ is the original feature of node i, and $\mathbf{h}_i \in \mathbb{R}^d$ is the projected feature of node i. $\mathbf{W}_{A'} \in \mathbb{R}^{d \times d_{A'}}$ is the learnable mapping matrix for type A', $\mathbf{b}_{A'} \in \mathbb{R}^d$ is the learnable bias vector, and $\sigma(\cdot)$ is an activation function, respectively.

2) Intra-Graph Aggregation: Given a metapath-based graph $\mathcal{G}^\Phi$, for each node i, we collect its metapath-based neighbors $\mathcal{N}_i^\Phi$ from $\mathcal{G}^\Phi$ and aggregate the embeddings of nodes in $\mathcal{N}_i^\Phi$. As the metapath-based neighbors exhibit different degrees of importance to the target node in contrastive tasks due to their distinct features, it is appropriate to assign different weights to them. Furthermore, our Metapath-based Neighbor Mask results in different metapath-based neighbors for each node throughout the training process, suggesting that our aggregation method should not rely on the specific topology. Thus, we adopt a graph attention layer [43] to aggregate the embeddings of nodes in $\mathcal{N}_i^\Phi$, which leverages a self-attention mechanism to assign different weights to neighbors, effectively and flexibly capturing the importance of each neighbor.

Specifically, for node i, the importance of its metapath-based neighbor node j is calculated as:

$$e_{ij}^\Phi = \text{LeakyReLU}\left(\mathbf{a}^{\Phi^\top} \cdot [\mathbf{h}_i \oplus \mathbf{h}_j]\right), \tag{3}$$

where $\mathbf{a}^\Phi \in \mathbb{R}^{2\,d}$ is the intra-graph attention vector for metapath Φ and $\oplus$ denotes the concatenate operation. After obtaining the importance of all the metapath-based neighbors for node i, we normalize them to get the weight coefficient and compute the weighted combination of the representations for node i:

$$\alpha_{ij}^\Phi = \frac{\exp\left(e_{ij}^\Phi\right)}{\sum_{s \in \mathcal{N}_i^\Phi} \exp\left(e_{is}^\Phi\right)}, \tag{4}$$

$$\mathbf{z}_i^\Phi = \text{PReLU}\left(\sum_{j \in \mathcal{N}_i^\Phi} \alpha_{ij}^\Phi \cdot \mathbf{h}_j + \mathbf{b}^\Phi\right). \tag{5}$$

Here $\mathbf{b}^\Phi \in \mathbb{R}^d$ is the bias vector for aggregation in metapath Φ, and finally the output embedding goes through an activation function $\text{PReLU}(\cdot)$.

To stabilize the learning process of intra-graph attention, and enrich the representation of embedding, we extend the attention mechanism to multiple heads. Specifically, K independent attention mechanisms are executed, and then these output embeddings are concatenated, resulting in the following representation:

$$\mathbf{z}_i^\Phi = \bigoplus_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i^\Phi} \left[\alpha_{ij}^\Phi\right]_k \cdot \mathbf{W}_k \mathbf{h}_i + \mathbf{b}^\Phi\right), \tag{6}$$

where $\mathbf{W}_k \in \mathbb{R}^{\frac{d}{K} \times d}$ is the input linear transformation matrix for each head, which is used to extract different node features and keep the dimension of $\mathbf{z}_i^\Phi$ to be d. And $[\alpha_{ij}^\Phi]_k$ is the normalized importance of node j to node i at the k-th attention head.

3) Inter-Graph Aggregation: After obtaining the node embeddings within each metapath-based graph, we need to combine these embeddings to obtain final node embeddings. Similar to the above discussion, different metapaths carry different semantic information, leading to different importance for contrastive learning tasks. Additionally, the Metapath-based Graph Mask results in different input metapath-based graphs throughout the training process. Therefore, we employ graph-level attention to automatically learn the significance of different metapaths and fuse the corresponding node embeddings with the learned weights for contrastive tasks.

To be specific, for each metapath-based graph $\widetilde{\mathcal{G}^\Phi} \in \widetilde{\mathcal{S}_A}$, we obtain a summary vector for graph $\widetilde{\mathcal{G}^\Phi}$ by averaging the transformed $\widetilde{\mathcal{G}^\Phi}$-specific node embeddings for all nodes $i \in \mathcal{V}_A$:

$$s^\Phi = \frac{1}{|\mathcal{V}_A|} \sum_{i \in \mathcal{V}_A} \tanh\left(\mathbf{W}_A \cdot \mathbf{z}_i^\Phi + \mathbf{b}_A\right), \tag{7}$$

where $\mathbf{W}_A \in \mathbb{R}^{d \times d}$ is the learnable weight matrix, $\mathbf{b}_A \in \mathbb{R}^d$ is the learnable bias vector.

Then we utilize a learnable vector $\mathbf{q}_A \in \mathbb{R}^d$ to compute the importance of each metapath-based graph:

$$e^\Phi = \mathbf{q}_A^\top \cdot s^\Phi, \tag{8}$$

Finally, we normalize the importance score to get the weight coefficient via softmax function, and aggregate all the $\widetilde{\mathcal{G}^\Phi}$-specific node embeddings for each node to obtain final node embedding:

$$\beta^\Phi = \frac{\exp\left(e^\Phi\right)}{\sum_{p=1}^{|\widetilde{\mathcal{S}_A}|} \exp\left(e^{\Phi_p}\right)}, \tag{9}$$

$$\mathbf{z_i} = \sum_{p=1}^{|\widetilde{\mathcal{S}_A}|} \beta^{\Phi_p} \cdot \mathbf{z}_i^{\Phi_p}. \tag{10}$$

E. Contrastive Optimization

As described in Section IV-A, to perform contrastive learning, we generate two augmented views of heterogeneous graph with the augmentation schemes described in Section IV-C, and obtain final node embeddings in these two views with the graph encoder. For each node i, we denote its final node embeddings in these two views as $\mathbf{z}_i^1$ and $\mathbf{z}_i^2$.

Before calculating the contrastive objective, we feed these embeddings into a multi-layer perceptron (MLP) with one hidden layer, since adopting nonlinear projection before calculating the contrastive objective is beneficial to improving the quality of representations [31]:

$$\begin{aligned} \mathbf{z}_i^1\_proj &= \mathbf{W}^{(2)}\sigma\left(\mathbf{W}^{(1)}\mathbf{z}_i^\alpha + \mathbf{b}^{(1)}\right) + \mathbf{b}^{(2)}, \\ \mathbf{z}_i^2\_proj &= \mathbf{W}^{(2)}\sigma\left(\mathbf{W}^{(1)}\mathbf{z}_i^\beta + \mathbf{b}^{(1)}\right) + \mathbf{b}^{(2)}, \end{aligned} \tag{11}$$

After that, we employ a contrastive objective, i.e. a discriminator, that distinguishes the embeddings of the same node in these two different views from other node embeddings. Specifically, we adapt the InfoNCE objective [44] to our multi-view graph contrastive learning setting. For each node i, we have the following

Algorithm 2: The HGCMA Training Algorithm.

Input: The heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,
node types $\mathcal{A} = \{A_1, A_2, \ldots, A_{|\mathcal{A}|}\}$,
target node type A,
metapaths $\mathcal{P}_A = \{\Phi_1, \Phi_2, \ldots \Phi_{P_A}\}$
Output: Learned parameters of graph encoder.
Construct the extended graph view $\mathcal{S}_A = \{\mathcal{G}^{\Phi_1}, \mathcal{G}^{\Phi_2}, \ldots \mathcal{G}^{\Phi_{P_A}}\}$ from $\mathcal{G}$ and $\mathcal{P}_A$
for epoch $\leftarrow 1, 2, \ldots$ **do**
  Generate two augmented views with `Augmentation process`;
  Obtain node embeddings of the two views with (2)–(10);
  Compute the contrastive objective $\mathcal{J}$ with (13);
  Update parameters by applying stochastic gradient ascent to maximize $\mathcal{J}$.
end for

objective under the first view:

$$\mathcal{J}_i^1 = \log \frac{\exp\left(\text{sim}\left(\mathbf{z}_i^1\_proj, \mathbf{z}_i^2\_proj\right)/\tau\right)}{\sum_{j \in \mathcal{V}_A} \exp\left(\text{sim}\left(\mathbf{z}_i^1\_proj, \mathbf{z}_j^2\_proj\right)/\tau\right)}, \quad (12)$$

where $\text{sim}(\cdot, \cdot)$ denotes the cosine similarity between two vectors, and τ denotes a temperature parameter. Since two views are symmetric, the objective $\mathcal{J}_i^2$ is computed similarly as $\mathcal{J}_i^1$.

Therefore, the overall objective to be maximized is defined as the average over all nodes in both views, given by:

$$\mathcal{J} = \frac{1}{2|\mathcal{V}_A|} \sum_{i \in \mathcal{V}_A} \left[\mathcal{J}_i^1 + \mathcal{J}_i^2\right]. \quad (13)$$

To sum up, in each training epoch, we first generate two graph views based on two mask mechanisms, and then obtain the final node embeddings of the two views by the encoder as described in Section IV-D. Finally, the model parameters are updated by maximizing the objective in (13). The entire training procedure is summarized in Algorithm 2. After training, to capture the full information in heterogeneous graph, we obtain the representations of nodes without mask mechanism.

F. Computational Complexity Analysis

According to Algorithm 2, there are three main phases in the training process: 1) data augmentation, 2) graph encoding, and 3) contrastive optimization.

In data augmentation, the computational complexity for Metapath-based Graph Mask is $O(|\mathcal{P}_A|)$. In Metapath-based Neighbor Mask, when $\alpha_n \geq 0.5$, data augmentation can be performed by selecting edges and constructing a new graph; when $\alpha_n < 0.5$, data augmentation can be performed by selecting edges and deleting them from original graph. Thus, its computational complexity is $O(\min(\alpha_n, 1-\alpha_n)|\mathcal{P}_A||\mathcal{E}_A|)$, where $|\mathcal{E}_A|$ is the average number of edges in $\mathcal{S}_A$.

In graph encoding process, the computational complexities for feature transformation, intra-graph aggregation and inter-graph aggregation are $O(d^2 \sum_{q \in Q} |\mathcal{V}_q|)$, $O(|\mathcal{P}_A|(d^2|\mathcal{E}_A| + d^2|\mathcal{V}_A|))$ and $O(d^2|\mathcal{P}_A||\mathcal{V}_A|)$, respectively, where Q is the actual participated node types.

In contrastive optimization, the computational complexity for (13) is $O(d^2|\mathcal{V}_A|^2)$. Thus, the calculation of the objective costs the most computational time during the training stage. For most existing work [13], [14], [15], [16], the computational complexity for contrastive objective in each metapath-based graph is also $O(d^2|\mathcal{V}_A|^2)$. Thus, their total computational complexity for contrastive objective is $O(d^2|\mathcal{P}_A||\mathcal{V}_A|^2)$.

TABLE I
STATISTICS OF THE DATASETS

Dataset	Node	Relation	Metapath
ACM	paper (P):3025 author (A):5912 subject (S):57	P-A:9936 P-S:3025	PAP PSP
DBLP	author (A):4057 paper (P):14328 conference (C):20	P-A:19645 P-C:14328	APA APCPA
IMDB	movie (M):4661 actor (A):5841 direct (D):2270	M-A:13983 M-D:4661	MAM MDM

V. EXPERIMENTS

In this section, we conduct experiments on three real-world datasets. The experiments aim to address the following research questions: **(RQ1)** Does our proposed HGCMA outperform state-of-the-art baselines on node classification task? **(RQ2)** How does HGCMA perform in clustering nodes? **(RQ3)** How does HGCMA perform in qualitative evaluation (visualization)? **(RQ4)** What is the impact of the augmentation schemes on the performance of our HGCMA? **(RQ5)** Is the proposed model sensitive to hyperparameters? How do key hyperparameters, i.e., α_g and α_n, impact the model performance? **(RQ6)** How does HGCMA perform in different metapath combinatons? **(RQ7)** Is the real runtime of HGCMA shorter than that of the baselines following multi-graph paradigm?

A. Experimental Setup

Datasets To evaluate our model, we use three publicly available real-world heterogeneous graph datasets[2] [45], [46], [47], the statistics of which are provided in Table I.

- *ACM* [5]: The target nodes are papers, which are divided into three classes: database, data mining, and wireless communication. Each paper has an average of 3.28 authors and one subject.
- *DBLP* [17]: The target nodes are authors, which are divided into four classes: database, data mining, machine learning, and information retrieval. Each author has an average of 4.84 papers.
- *IMDB* [5]: The target nodes are movies, which are divided into three classes: action, comedy, and drama. Each movie has an average of 3 actors and one director.

[2]The data is available for download at https://github.com/pkuliyi2015/GraphMSE/tree/main/data

Baselines To comprehensively evaluate our model, we compare HGCMA with several state-of-the-art baselines that incorporate diverse modeling and training approaches. As HGCMA is a contrastive representation learning method for heterogeneous graphs, to directly evaluate HGCMA in this research direction, we select **DMGI** [13], **HeCo** [12], and **HGCML** [16] as baselines. Additionally, to compare with traditional representation learning methods for heterogeneous graphs, we select **Mp2vec** [4] as a baseline. Since most research on graph contrastive learning focuses on homogeneous graphs, we also include **MVGRL** [36], **DGI** [34], and **GraphCL** [10] as baselines to compare HGCMA with state-of-the-art methods for graph contrastive learning. Finally, as HGCMA is an unsupervised method, to assess its competitiveness in downstream tasks, we compare it with semi-supervised methods, including **HAN** [5], **MAGNN** [23], **GTN** [45] and **GCN** [3]. The former three methods are designed for heterogeneous graphs, while GCN is designed for homogeneous graphs.

For all baselines except GCN, we conduct experiments based on their official implementations.

Implementation Details For our proposed HGCMA, we initialize model parameters using Xavier initialization [48], and train the model with Adam optimizer [49]. We search the learning rate from 1e-4 to 5e-3, and tune the epoch number from 100 to 1,200 with a step size of 50. For the dropout probability, it is selected from 0.1 to 0.7 with step size 0.05. τ is tuned from 0.4 to 0.9 with a step size of 0.1, α_g is tuned from 0.1 to 1.0 with a step size of 0.1, and α_n is tuned from 0.1 to 0.9 with a step size of 0.1.

For baselines, the unsupervised models are trained following the settings in their original papers, the semi-supervised methods are optimized through an end-to-end supervised manner, and the homogeneous models treat all nodes in the graph as the same type. For each dataset, we tune hyperparameters by grid search.

For all methods, we set the embedding dimension to 64 and randomly run five times and report the average results. For a fair comparison, we use the same metapath sets for all methods which utilize metapaths. And we keep the same metapath setting with previous works [5], [47], as shown in Table I.

Evaluation Metrics We quantitatively evaluate the performance of models on node classification and node clustering tasks. For the node classification task, we evaluate models' performance using Macro-F1 (Ma-F1) and Micro-F1 (Mi-F1) scores. The F1 score is the harmonic mean between precision and recall in binary classification. Macro-F1 is computed as the arithmetic mean of all per-class F1 scores, while Micro-F1 calculates a global average F1 score by considering the sums of True Positives (TP), False Negatives (FN), and False Positives (FP). For the node clustering task, we evaluate models using Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) [50]. NMI measures the normalized mutual information between the ground truth partition and the cluster assignments. In contrast, ARI is the corrected-for-chance version of the Rand Index, which quantifies the similarity between the ground truth partition and the cluster assignments.

B. Node Classification (RQ1)

To begin with, we answer (RQ1) by evaluating the performance of HGCMA on node classification task on the three datasets. For each unsupervised method, we follow the linear evaluation protocol introduced in [34], where each model is firstly trained in an unsupervised manner; then, the resulting embeddings are used to train and test a simple l_2-regularized logistic regression classifier. Each set of embeddings is tested 10 times. For each dataset, in order to simulate scenarios where labels are scarce, the percentage of training labeled nodes are set as 1%, 3%, and 5%, and the rest of the labeled nodes are split into 30% validation set and 70% testing set.

The results are shown in Table II. As can be seen, the proposed HGCMA outperforms all the methods on all datasets and all splits, and have considerable performances even though the percentage of training labeled nodes is only 1%. It verifies the superiority of our HGCMA.

We make other observations as follows. Firstly, all the heterogeneous graph contrastive learning methods (DMGI, HeCo, HGCML, and HGCMA) achieve better performance than homogeneous graph contrastive learning methods (MVGRL, DGI, and GraphCL) in most cases. It indicates the importance of utilizing metapath to extend original graph data for heterogeneous graph representation learning. Moreover, HGCMA achieves the best performance among the heterogeneous graph contrastive learning methods. It further verifies the effectiveness of the two masking augmentation schemes and the directly contrastive optimization method.

Secondly, for semi-supervised methods, their performances are not the best among baselines in most cases. We conjecture that it is caused by the lack of training data (contrasting 1%/3%/5% with 20%/40%/60%/80% training data [5], [23]). And we can find that in the cases of 1% training data, the performances of semi-supervised methods are very unstable, which corroborates our speculation again.

C. Node Clustering (RQ2)

Subsequently, we answer (RQ2) by comparing the performance of unsupervised methods on node clustering task. We utilize K-means algorithm [51] to cluster the learned embeddings of all labeled nodes, and set the number of clusters as the number of classes on each dataset. To alleviate the instability due to different initial values, we repeat each experiment 10 times, and report the average results, which are shown in Table III. We can see that HGCMA consistently achieves the best results on all the datasets, which proves the effectiveness of HGCMA again. Also, heterogeneous graph contrastive learning methods usually achieve much better performance than homogeneous graph contrastive learning methods. It verifies the importance of metapath-based graphs for heterogeneous graph representation learning once more.

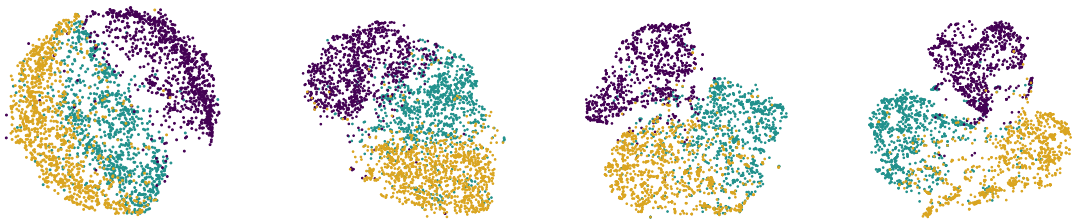
D. Visualization (RQ3)

To further evaluate the qualities of embeddings, we visualize the learned embeddings on ACM dataset. Specifically, we plot

TABLE II
QUANTITATIVE RESULTS (%±σ) ON NODE CLASSIFICATION

Datasets	Metric	Split	MVGRL	DGI	GraphCL	DMGI	HeCo	HGCML	Mp2vec	HAN	MAGNN	GTN	GCN	HGCMA
ACM	Ma-F1	1%	82.00±5.0	87.25±1.5	87.26±1.4	87.89±3.5	88.13±2.0	88.09±0.6	61.61±3.7	82.37±8.7	77.72±5.8	74.83±10.1	66.30±9.6	**89.61±2.2***
		3%	86.27±1.5	88.67±1.0	88.74±1.0	89.94±0.8	89.42±0.6	90.00±0.5	68.29±1.3	90.10±0.8	85.93±1.0	88.75±1.3	87.62±1.3	**91.26±0.6***
		5%	87.22±1.5	89.44±0.7	89.42±0.9	90.31±0.8	89.72±0.6	90.37±0.6	69.09±1.5	91.00±0.5	87.59±1.5	90.13±1.7	89.22±3.4	**91.30±0.7**
	Mi-F1	1%	82.03±4.8	87.20±1.5	87.19±1.4	87.95±3.4	88.02±2.0	88.10±0.6	62.19±3.4	83.31±7.0	78.17±4.8	76.14±8.0	68.61±7.2	**89.54±2.2***
		3%	85.98±1.6	88.62±1.0	88.68±1.0	89.85±0.8	89.27±0.6	89.81±0.5	68.25±1.3	90.05±0.8	85.92±0.9	88.72±1.2	87.54±1.3	**91.14±0.7***
		5%	87.07±1.5	89.43±0.7	89.47±1.0	90.20±0.8	89.63±0.6	90.32±0.6	69.05±1.4	90.98±0.5	87.56±1.5	90.03±1.8	89.22±3.4	**91.21±0.7**
DBLP	Ma-F1	1%	81.79±4.4	79.75±2.3	76.94±3.9	86.05±2.2	87.36±6.8	90.49±0.4	72.34±6.0	86.15±8.2	87.41±7.8	69.58±7.7	75.99±8.2	**91.10±2.9**
		3%	85.37±1.3	80.75±1.2	79.81±1.0	89.03±1.0	90.77±0.7	90.76±0.6	82.11±1.7	90.11±1.2	92.16±0.5	80.03±1.4	85.53±1.2	**92.80±0.4***
		5%	86.08±0.8	81.56±0.7	80.21±0.8	89.48±0.9	90.76±0.6	91.36±0.3	84.36±1.0	89.74±1.5	92.48±0.6	82.54±1.2	87.29±0.6	**92.96±0.4**
	Mi-F1	1%	83.22±2.7	80.63±1.9	78.06±3.0	87.34±1.6	89.20±3.4	91.08±0.4	75.09±3.6	88.58±4.6	89.69±4.2	73.33±4.5	79.04±4.4	**91.93±1.8**
		3%	86.14±1.2	81.53±1.1	80.44±0.9	89.77±0.9	91.39±0.7	91.30±0.5	83.45±1.7	90.86±1.1	92.81±0.4	81.14±1.4	86.39±1.2	**93.27±0.4***
		5%	86.74±0.8	82.22±0.6	80.84±0.9	90.19±0.8	91.34±0.6	91.99±0.3	85.33±0.9	90.54±1.3	93.07±0.5	83.27±1.2	88.02±0.6	**93.40±0.4**
IMDB	Ma-F1	1%	33.53±2.3	40.95±4.5	38.97±4.1	45.37±4.2	42.11±3.8	43.26±1.0	33.81±1.4	25.20±4.4	35.76±3.8	36.48±5.1	33.20±9.7	**48.84±5.4***
		3%	38.04±2.4	47.70±2.8	45.51±3.1	50.07±2.5	49.60±2.1	46.71±2.2	35.44±1.1	36.96±6.2	47.56±2.7	44.31±3.0	42.51±7.4	**55.74±2.4***
		5%	38.63±2.5	49.74±2.3	47.75±2.8	53.09±2.3	50.03±2.3	49.85±2.6	35.67±1.0	47.48±6.1	50.05±2.7	45.78±2.9	41.26±8.8	**57.54±2.0***
	Mi-F1	1%	47.97±2.4	49.75±2.1	49.46±2.1	47.74±3.8	53.35±3.1	51.40±1.2	44.46±2.9	49.52±1.9	49.97±1.3	50.37±1.0	51.62±3.9	**57.97±3.2***
		3%	48.69±2.8	51.59±2.3	50.41±2.5	51.28±2.3	56.61±2.0	53.28±2.3	45.05±2.7	53.03±1.6	53.04±2.0	52.16±2.6	53.33±2.9	**60.00±1.9***
		5%	49.66±1.1	53.78±2.0	52.15±2.2	54.79±2.3	57.76±1.2	55.79±2.3	s46.76±1.4	57.06±1.6	56.40±1.1	52.27±2.3	55.86±3.1	**62.37±1.6***

The best and second-best runs per case are indicated with boldface and underline, respectively. Scores marked with * denote that the improvement is statistically significant compared with the best baseline (two-tailed independent t-test with p-value < 0.05).



(a) MVGRL (b) DGI (c) HeCo (d) HGCMA

Fig. 4. 2-D visualization of the inferred embeddings on the ACM dataset. The same color indicates the same class label.

TABLE III
QUANTITATIVE RESULTS (%) ON NODE CLUSTERING

Datasets	ACM		DBLP		IMDB	
Metrics	NMI	ARI	NMI	ARI	NMI	ARI
MVGRL	32.17	29.01	20.82	15.96	0.34	-0.91
DGI	55.74	58.79	16.55	11.22	1.50	-0.23
GraphCL	21.66	17.31	46.49	47.19	2.01s	0.68
DMGI	53.29	50.39	68.26	74.86	7.14	4.62
HeCo	60.58	63.14	71.92	77.38	4.50	4.28
HGCML	53.58	51.92	71.18	76.94	9.24	6.70
Mp2vec	21.65	15.78	34.78	42.19	0.81	-0.09
HGCMA	**69.95***	**74.40***	**76.61***	**81.45***	**10.29***	**9.44***

Scores marked with * denote that the improvement is statistically significant compared with the best baseline (two-tailed independent t-test with p-value < 0.05).

TABLE IV
COMPARISON OF HGCMA AND ITS VARIANTS

Datasets	ACM		DBLP		IMDB	
Metrics	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1
HGCMA$_{no-mask}$	84.79	84.85	60.95	62.19	40.54	49.67
HGCMA$_{g-mask}$	89.86	89.74	91.85	92.30	49.50	54.53
HGCMA$_{n-mask}$	90.98	90.86	91.91	92.48	52.81	57.32
HGCMA	**91.26**	**91.14**	**92.80**	**93.27**	**55.74**	**60.00**

the embeddings of paper nodes in MVGRL, DGI, HeCo and HGCMA using t-SNE [52], and the results are shown in Fig. 4, where different colors indicate different labels. We can find that although all of the baselines present relatively clear boundaries, HGCMA still achieves the most separated clusters compared with the baseline methods. In particular, HGCMA separates the green cluster from the yellow cluster further apart. This result can also explain why our approach achieves better performance on the previous two tasks.

E. Ablation Study (RQ4)

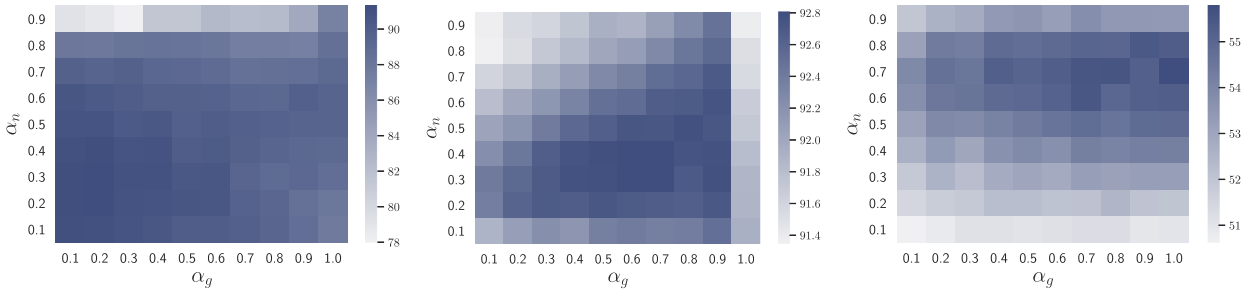
In order to verify the effectiveness of our augmentation schemes, we design three variants of HGCMA:

- HGCMA$_{no-mask}$: This model does not perform any augmentations in the training process.
- HGCMA$_{g-mask}$: This model only performs metapath-based graph masking in the training process.
- HGCMA$_{n-mask}$: This model only performs metapath-based neighbor masking in the training process.

We report the results of our method on data splits with 3% training data, which are shown in Table IV. We can see that both augmentation schemes improve model performance significantly on all datasets, indicating the necessity of data augmentation for heterogeneous graph contrastive learning. In addition, HGCMA achieves better results compared with HGCMA$_{n-mask}$, and it verifies the effectiveness of leveraging the information of the whole extended view in augmentation scheme.

F. Analysis of Hyper-Parameters (RQ5)

In this section, we perform sensitivity analysis on two main hyperparameters in HGCMA: α_g and α_n. We conduct node classification on data splits with 3% training data and report the average Macro-F1 values (a similar tendency is observed in Micro-F1 and other percentages). The results on three datasets

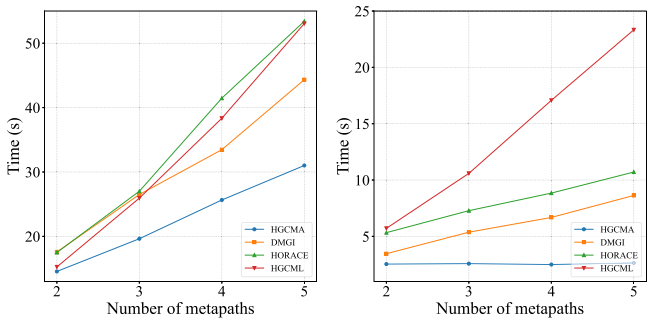


(a) ACM (b) DBLP (c) IMDB

Fig. 5. Performance of HGCMA with different α_g and α_n on the three datasets (data splits with 5% training data) in terms of Macro-F1 values. The optimal (α_g, α_n) pairs are $(0.2, 0.3)$, $(0.7, 0.3)$ and $(1.0, 0.7)$, respectively.

TABLE V
NODE CLASSIFICATION PERFORMANCE OF DMGI, HECO AND HGCMA UNDER DIFFERENT METAPATH COMBINATIONS

meta-paths	Metric	Split	DMGI	HeCo	HGCML	HGCMA
PAP PSP	Ma-F1	1%	87.89	88.13	88.09	**89.61**
		3%	89.94	89.42	90.00	**91.26**
		5%	90.31	89.72	90.37	**91.30**
	Mi-F1	1%	87.95	88.02	88.10	**89.54**
		3%	89.85	89.27	89.81	**91.14**
		5%	90.20	89.63	90.32	**91.21**
PAP PSP PAPSP	Ma-F1	1%	86.21	87.78	81.30	**88.39**
		3%	89.23	88.32	86.02	**91.23**
		5%	89.93	89.26	88.32	**91.26**
	Mi-F1	1%	86.35	87.68	81.43	**88.36**
		3%	89.09	88.31	85.72	**91.15**
		5%	89.73	89.41	88.24	**91.19**
PAP PSP PAPSP PAPAP	Ma-F1	1%	86.01	86.34	79.62	**86.77**
		3%	88.38	88.17	84.74	**91.22**
		5%	89.42	89.13	86.26	**91.24**
	Mi-F1	1%	86.06	86.24	79.89	**87.14**
		3%	87.90	88.13	84.40	**91.24**
		5%	89.36	89.15	86.29	**91.26**
PAPSP PAPAP	Ma-F1	1%	61.82	71.50	55.69	**81.55**
		3%	68.09	77.82	68.31	**82.55**
		5%	71.63	77.86	70.47	**82.36**
	Mi-F1	1%	63.40	71.29	58.25	**81.67**
		3%	68.37	78.09	68.55	**82.47**
		5%	72.37	78.09	70.79	**82.61**



(a) Full runtime. (b) Contrastive loss computation.

Fig. 6. Average runtimes for 100 training epochs according to the number of metapaths on ACM dataset.

are shown in Fig. 5. With the help of the values in the colorbar, we can see that HGCMA has a relatively good performance in most hyperparameter combinations. The performance may degrade only when the α_g or α_n takes marginal values. We thus conclude that, overall, our augmentation schemes are insensitive to these hyperparameters, demonstrating their robustness. We can also find that the optimal hyperparameters for the three datasets are quite different: in ACM, the optimal α_g and α_n are small; in DBLP, the optimal α_g is large and the optimal α_n is small; an in IMDB, the optimal α_g and α_n are large. It indicates that in different heterogeneous graphs, we need different levels of perturbation to perform contrastive learning, and our proposed HGCMA can perform data augmentation with controllable perturbation to adapt to different heterogeneous graphs.

G. Analysis of Metapath Selection (RQ6)

To explore the effect of the choice of metapaths, we conduct node classification experiments on different metapath combinations on ACM dataset. The used metapath combinations and results of DMGI, HeCo, HGCML, and HGCMA are shown in Table V.

We find that, with PAPSP and PAPAP added to the metapath combination, the performance of DMGI, HeCo, and HGCML are gradually declining, which indicates that when inappropriate metapath is involved, simply fusing corresponding node embeddings with predefined weights would affect the distinctiveness of node embeddings; The performances of HGCMA are only decreased on 1% split and maintain stable on 3% and 5% split. It proves that our directly contrastive optimization method could help utilize metapaths efficiently and robustly. Besides, when only using PAPSP and PAPAP, all the methods' performances experience a decline. However, HGCMA still gets the best performance by a larger margin. This comparison verifies the superiority of our directly contrastive optimization method again.

H. Runtime Comparison (RQ7)

To compare the runtime of HGCMA with baselines following multi-graph paradigm, we reproduce DMGI, HORACE [15] and HGCML using the same implementation framework as that of HGCMA. Experiments are conducted on a server equipped with an Intel Xeon E5-2680 v4 14-Core CPU and an Nvidia RTX 2080Ti GPU. Fig. 6(a) illustrates the average full runtimes for 100 training epochs of each method under different numbers of

metapaths on ACM dataset, and Fig. 6(b) illustrates the average contrastive loss computation runtimes. The results demonstrate that HGCMA has the shortest runtime and the slowest growth rate as the number of metapaths increases. This is because the computational burden for the contrastive objective in HGCMA remains constant, regardless of the number of metapaths.

VI. Conclusion

In this article, we propose a **H**eterogeneous **G**raph **C**ontrastive learning model with **M**etapath-based **A**ugmentations (**HGCMA**), which is designed for downstream tasks with a small amount of labeled data. HGCMA utilizes metapaths to extend the original graph data by constructing metapath-based graphs. And based on the extended view, it constructs different augmented views by masking metapath-based graph and edges. Then, a two-stage attention-based graph encoder is leveraged to output the final node embeddings and the parameters of our model are learned by optimizing the contrastive loss of the final embeddings. With the proposed augmentations, our HGCMA is able to fully leverage the information of metapath-based graphs. And with the directly contrastive optimization method, our HGCMA could learn discriminative node representations with low computational complexity. Extensive experiments on three public datasets validate the effectiveness of HGCMA compared with state-of-the-art methods.

In future work, we will investigate how to automatically select useful metapaths in the contrastive learning process.

References

[1] Y. Sun and J. Han, "Mining heterogeneous information networks: A structural analysis approach," in *Proc. ACM SIGKDD Explorations Newslett.*, 2012, pp. 20–28.

[2] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[3] N. T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–14.

[4] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 135–144.

[5] X. Wang et al., "Heterogeneous graph attention network," in *Proc. World Wide Web Conf.*, 2019, pp. 2022–2032.

[6] P. Yu, C. Fu, Y. Yu, C. Huang, Z. Zhao, and J. Dong, "Multiplex heterogeneous graph convolutional network," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2022, pp. 2377–2387.

[7] C. Shi, B. Hu, X. W. Zhao, and S. P. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.

[8] B. Hu, C. Shi, X. W. Zhao, and S. P. Yu, "Leveraging meta-path based context for top- n recommendation with a neural co-attention model," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1531–1540.

[9] A. El-Kishky et al., "TwHIN: Embedding the twitter heterogeneous information network for personalized recommendation," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2022, pp. 2842–2850.

[10] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 5812–5823.

[11] J. Xia, L. Wu, J. Chen, B. Hu, and S. Z. Li, "SimGRACE: A simple framework for graph contrastive learning without data augmentation," in *Proc. World Wide Web Conf.*, 2022, pp. 1070–1079.

[12] X. Wang, N. Liu, H. Han, and C. Shi, "Self-supervised heterogeneous graph neural network with co-contrastive learning," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 1726–1736.

[13] C. Park, D. Kim, J. Han, and H. Yu, "Unsupervised attributed multiplex network embedding," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 5371–5378.

[14] B. Jing, S. Feng, Y. Xiang, X. Chen, Y. Chen, and H. Tong, "X-GOAL: Multiplex heterogeneous graph prototypical contrastive learning," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 894–904.

[15] Y. Zhu, Y. Xu, H. Cui, C. Yang, Q. Liu, and S. Wu, "Structure-enhanced heterogeneous graph contrastive learning," in *Proc. SIAM Int. Conf. Data Mining*, 2022, pp. 82–90.

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

[17] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[18] C. Yang, Y. Xiao, Y. Zhang, Y. Sun, and J. Han, "Heterogeneous network representation learning: A unified framework with survey and benchmark," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4854–4873, Oct. 2022.

[19] Y. Ren, B. Liu, C. Huang, P. Dai, L. Bo, and J. Zhang, "HDGI: An unsupervised graph neural network for representation learning in heterogeneous graph," in *Proc. AAAI Conf. Artif. Intell. Workshop*, 2020, pp. 1–6.

[20] Y. Shi, Q. Zhu, F. Guo, C. Zhang, and J. Han, "Easing embedding learning by comprehensive transcription of heterogeneous information networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2190–2199.

[21] H. Chen, H. Yin, W. Wang, H. Wang, Q. V. H. Nguyen, and X. Li, "PME: Projected metric embedding on heterogeneous networks for link prediction," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1177–1186.

[22] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proc. 15th Eur. Semantic Web Conf.*, 2018, pp. 593–607.

[23] F. Xinyu, Z. Jiani, M. Ziqiao, and K. Irwin, "MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proc. World Wide Web Conf.*, 2020, pp. 2331–2341.

[24] T.-y. Fu, W.-C. Lee, and Z. Lei, "HIN2Vec: Explore meta-paths in heterogeneous information networks for representation learning," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2017, pp. 1797–1806.

[25] Y. He, Y. Song, J. Li, C. Ji, J. Peng, and H. Peng, "HeteSpaceyWalk: A heterogeneous spacey random walk for heterogeneous information network embedding," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 639–648.

[26] J. Tang, M. Qu, and Q. Mei, "PTE: Predictive text embedding through large-scale heterogeneous text networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 1165–1174.

[27] W. Zhang, Y. Fang, Z. Liu, M. Wu, and X. Zhang, "mg2vec: Learning relationship-preserving heterogeneous graph representations via meta-graph embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1317–1329, Mar. 2022.

[28] J. Li et al., "Higher-order attribute-enhancing heterogeneous graph neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 560–574, Jan. 2023.

[29] H. Ji, X. Wang, C. Shi, B. Wang, and P. S. Yu, "Heterogeneous graph propagation network," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 521–532, Jan. 2023.

[30] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2019, pp. 15509–15519.

[31] C. Ting, K. Simon, N. Mohammad, and H. Geoffrey, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[32] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 776–794.

[33] L. Wu, H. Lin, C. Tan, Z. Gao, and S. Z. Li, "Self-supervised learning on graphs: Contrastive, generative, or predictive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 4216–4235, Apr. 2023.

[34] P. Velickovic, W. Fedus, L. W. Hamilton, P. Liò, Y. Bengio, and D. R. Hjelm, "Deep graph infomax," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–17.

[35] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–24.

[36] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4116–4126.

[37] X. Xu, C. Deng, Y. Xie, and S. Ji, "Group contrastive self-supervised learning on graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3169–3180, Mar. 2023.
[38] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3733–3742.
[39] Z. Yanqiao, X. Yichen, Y. Feng, L. Qiang, W. Shu, and W. Liang, "Deep graph contrastive representation learning," 2020, *arXiv:2006.04131*.
[40] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *Proc. World Wide Web Conf.*, 2021, pp. 2069–2080.
[41] Y. Zhang, H. Zhu, Z. Song, P. Koniusz, and I. King, "COSTA: Covariance-preserving feature augmentation for graph contrastive learning," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2022, pp. 2524–2534.
[42] X. Jiang, Y. Lu, Y. Fang, and C. Shi, "Contrastive pre-training of GNNs on heterogeneous graphs," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 803–812.
[43] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–12.
[44] A. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.
[45] S. Yun, M. Jeong, R. Kim, J. Kang, and J. H. Kim, "Graph transformer networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2019, pp. 11960–11970.
[46] Y. Li, Y. Jin, G. Song, Z. Zhu, C. Shi, and Y. Wang, "Graphmse: Efficient meta-path selection in semantically aligned feature space for graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 4206–4214.
[47] J. Zhao, Q. Wen, S. Sun, Y. Ye, and C. Zhang, "Multi-view self-supervised heterogeneous graph embedding," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2021, pp. 319–334.
[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
[49] P. D. Kingma and L. J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.
[50] V. X. Nguyen, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, 2010.
[51] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.
[52] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

Xiaoru Chen received the B.Sc. and M.Sc. degrees from Sun Yat-sen University, Guangzhou, China, in 2020 and 2023, respectively. His main research interests include the distributed graph neural networks and recommender systems.

Yingxu Wang received the B.Sc. degree from Sun Yat-sen University, Guangzhou, China, 2020, and the M.Sc. degree from Fudan University, Shanghai, China, in 2023. He is currently working toward the Ph.D. degree with the Mohammed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE. His main research interests include machine learning and AI for science.

Jinyuan Fang received the B.Sc. and M.Sc. degrees from Sun Yat-sen University, Guangzhou, China, in 2019 and 2022, respectively. He is currently working toward the Ph.D. degree with the University of Glasgow, Glasgow, U.K. His research interests include knowledge graphs and natural language processing.

Zaiqiao Meng received the Ph.D. degree from Sun Yat-sen University, Guangzhou, China. He is currently a Lecturer with the University of Glasgow, Glasgow, U.K., based within the Information Retrieval Group and School of Computing Science. Prior to that, he was a Postdoctoral Researcher with the Language Technology Lab, University of Cambridge, Cambridge, U.K., and a Postdoctoral Researcher with the Information Retrieval Group of the University of Glasgow. His research interests include information retrieval, graph neural networks, knowledge graphs, and natural language processing.

Shangsong Liang received the Ph.D. degree from the University of Amsterdam, Amsterdam, The Netherlands, in 2014. He is currently a Faculty Member with the Sun Yat-sen University, Guangzhou, China, and Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE. His expertise lies in the fields of machine learning, information retrieval and data mining. He was a (visiting) Postdoctoral Research Scientist with the University of Massachusetts Amherst, Amherst, MA, USA, and the University College London, London, U.K. He has extensively published his work in top-tier conferences and journals, including SIGIR, KDD, WWW, CIKM, AAAI, WSDM, NeurIPS, TKDE and TOIS. He was the recipient of an Outstanding Reviewer Award in SIGIR 2017.