




Semi-supervised heterogeneous graph contrastive learning with label-guided

Chao Li¹ · Guoyi Sun¹ · Xin Li¹ · Juan Shan² 

Accepted: 23 July 2024 / Published online: 3 August 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

Heterogeneous Graph Neural Networks represent a powerful approach to understand and utilize the intricate structures and semantics within complex graphs. When it comes to semi-supervised learning on graphs, the challenge lies in effectively leveraging labeled data to generalize predictions to unlabeled nodes. Traditional methods often fall short in fully utilizing labeled information, limiting their performance to the number of available labels. To overcome these limitations, in this paper, we propose a Semi-Supervised Heterogeneous Graph Contrastive Learning with Label-Guided (SSGCL-LG) model. SSGCL-LG tackles this challenge by fully integrating label information into the learning process through contrastive learning. Specifically, it constructs a label graph that incorporates both node and label representations, enhancing the supervised signal. Moreover, we propose a novel strategy for selecting positive and negative samples based on labels and meta-paths, effectively pulling positive samples closer together in the embedding space. To optimize node representations, SSGCL-LG combines contrastive loss with semi-supervised loss, enabling the model to learn from both labeled and unlabeled data. Extensive experiments on real-world datasets validate the effectiveness of our framework, demonstrating its superiority over existing methods. The code for this work is publicly available in the <https://github.com/sun281210/SSGCL-LG>.

Keywords Heterogeneous graph neural networks · Semi-supervised learning · Contrastive learning · Label information

1 Introduction

Heterogeneous Graph (HG), also known as Heterogeneous Information Network (HIN), are common network structures composed of multiple types of nodes and edges in the real world [1]. For example, an academic network can be represented as a HG, which consists of three types of nodes (author, paper, subject) and two types of edges (authors write papers,

papers contain subject) as shown in Fig. 1(a). Similarly, a network of legal documents, such as civil judgments, can also be represented as a HG, which consists of five types of nodes (plaintiff, defendant, judge, instrument, cause) and three types of edges (judges write instruments, instruments contain causes, plaintiffs and defendants are parties involved in the instruments) as shown in Fig. 1(b). In recent years, Heterogeneous Graph Neural Networks (HGNNs) have achieved significant success in handling HG data [2]. This is primarily due to their effective integration of message-passing mechanisms with the inherent complexity of heterogeneity, enabling a more comprehensive capture of the intricate structures and rich semantic information inherent in heterogeneous graphs [3]. With the prevalence of large-scale complex networks, HGNNs have become a powerful tool for processing fields such as social networks [4], e-commerce [5], smart justice [6], and bioinformatics [7].

Semi-supervised learning (SSL) [8] is a machine learning paradigm aimed at enhancing model performance by leveraging both labeled and unlabeled data. In traditional supervised learning, models are trained and predicted using only labeled data. However, in many practical scenarios, obtaining a large

✉ Juan Shan
skd992135@sdust.edu.cn

Chao Li
lichao@sdust.edu.cn

Guoyi Sun
1139457124@qq.com

Xin Li
2861649475@qq.com

¹ College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao 266590, Shandong, China

² College of Humanities and Law, Shandong University of Science and Technology, Qingdao 266590, Shandong, China

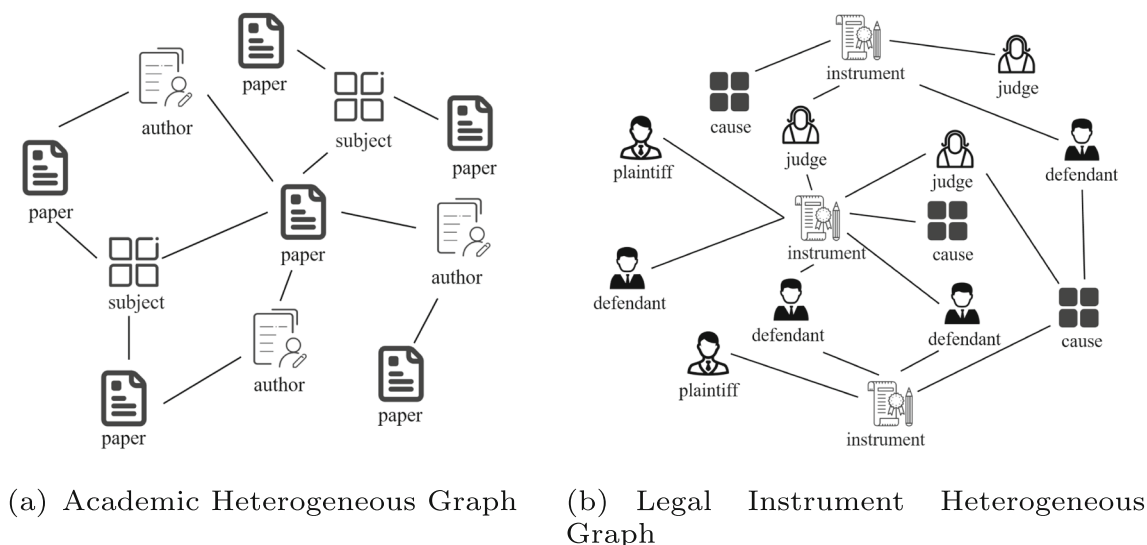


Fig. 1 Examples of heterogeneous graphs

amount of labeled data can be expensive or challenging. The goal of semi-supervised learning is to improve the generalization capability and performance of models by utilizing a limited amount of labeled data alongside a significant amount of unlabeled data.

In recent years, graph-based SSL methods have made significant progress [9]. However, few studies have provided an overarching view to address the core issue of SSL, which is that the insufficiency of labeled data can lead to overfitting and distribution shift problems in the model [10]. In addition, existing SSL methods such as GCN (Semi-supervised classification with graph convolutional network) [11], GraphSAGE (Inductive Representation Learning on Large Graphs) [12], GAT (Graph attention network) [13], etc., typically focus on learning the mapping function between node representation and labels, where labels are only used to compute the classification loss of the output. This means that the process of learning node representation does not fully utilize label information, limiting the comprehensive consideration of label information in SSL [14].

The fundamental idea of graph contrastive learning [15], a novel approach in self-supervised graph representation learning, aims to optimize the model by minimizing the distance between the target node and positive samples and maximizing the distance with negative samples [16]. Although contrastive learning can use the data itself to provide supervisory information for representation learning, it is not directly applicable to SSL [17]. Contrastive learning focuses on extracting features by learning the similarity and dissimilarity between data samples, typically used in unsupervised or self-supervised learning tasks. However, in semi-supervised learning, we often have a small amount of labeled data and a large amount of unlabeled data. In this scenario, contrastive

learning may face several challenges: firstly, due to the vast number of unlabeled data compared to labeled data, contrastive learning may be affected by issues such as excessive unlabeled data, difficulties in measuring similarity, leading to instability and inaccuracy in feature learning; secondly, semi-supervised learning emphasizes how to utilize information from labeled data to guide the learning process, while contrastive learning's core lies in unsupervised learning, which may not effectively utilize information from labeled data. Therefore, contrastive learning cannot be directly applied to semi-supervised learning. Furthermore, few studies have fully utilized valuable label information to supervise the construction of effective positive and negative samples in contrastive loss.

Indeed, labels can carry valuable information that is beneficial for node classification. Firstly, each label can be seen as a virtual center for nodes belonging to that label, reflecting the proximity of intra-class nodes. For example, in an academic network, papers within the same field are more relevant than those from different fields. In a business network, products within the same category often share similar characteristics. Secondly, labels are associated with rich semantics, and certain labels can be semantically close to each other. For instance, the fields of artificial intelligence and machine learning are more interrelated than artificial intelligence and chemistry. The relationship between computers and mice is closer than that between computers and digital cameras. Therefore, when classifying paper domains or product categories, it is essential to explore the rich information provided by labels. This motivates us to design a new framework that thoroughly considers the performance of GNNs in semi-supervised node classification by leveraging label information.

In this work, we focus on exploring, building upon, and proposing a label information based method for semi-supervised HGNN. To achieve this, we are faced with two key challenges: (1) How to explicitly incorporate label signals into the graph structure? (2) How to construct more reliable positive and negative samples by the label and semantic information in HGs?

To address the issues, we propose a new framework **Semi-Supervised Heterogeneous Graph Contrastive Learning with Label-Guided (SSGCL-LG)** designed to maximize the use of label information, thereby enhancing the performance of HGNNs in semi-supervised tasks. In the paper, we integrate rich label information comprehensively into GNN to facilitate semi-supervised node classification. We construct a label graph where a novel node is created for each label with semantic features, and connections are established with intra-class nodes, making each label act as the center for its corresponding nodes. By utilizing a message passing mechanism to jointly learn node and label representations, we can effectively smooth intra-class node representations and explicitly encode label semantics. Additionally, we apply label information to the selection of positive samples, fully leveraging label information to tightly integrate nodes of the same category in embeddings.

Specifically, to capture both homogeneous and heterogeneous neighborhood information effectively, we decompose the heterogeneous graph into multiple homogeneous and heterogeneous subgraphs based on metapaths. We first introduce a strategy where heterogeneous subgraphs guide the fusion of homogeneous subgraphs. Then, we treat labels as special nodes and design a label graph to explicitly encode label information into the learning process of Graph Neural Networks (GNNs). Furthermore, we introduce a contrastive loss for semi-supervised learning, aiming to fully leverage the supervisory signals inherent in the data itself. The semi-supervised contrastive loss is built upon the foundation of self-supervised contrastive loss functions, utilizing the supervision signals from both labeled and unlabeled data. This tightens the embedding of nodes within the same class, leading to improved classification accuracy. This method enables better utilization of label information in SSL, overcoming the challenge of sparse labeled data, thereby enhancing the performance of HGNNs in semi-supervised tasks.

The remainder of this paper is organized as follows: Section 2 surveys the related work. Section 3 presents some theories about heterogeneous graphs and provides formal definitions. Section 4 presents the Semi-Supervised Heterogeneous Graph Contrastive Learning with Label-Guided model. Section 5 describes the experiments performed in this study with results analysis. Finally, Section 6 concludes this paper and future work.

2 Related work

In this section, we will introduce the related work about graph neural networks, and give a brief description of graph representation learning as well as graph contrast learning.

2.1 Graph neural networks

GNNs propagate and aggregate node features through multiple neural layers to predict labels from feature propagation [18]. For example, GCN [11] obtains node representation that aggregate neighborhood information through an approximation of spectral graph convolutions. GAT [13] assigns attention coefficients based on the similarity of features between nodes to aggregate neighborhood information. GraphSAGE [12] samples a fixed number of neighbor nodes and aggregates the representation of neighbors at each layer. Additionally, AM-GCN (Adaptive Multi-channel Graph Convolutional Networks) [19] learns specific and common embedding for nodes in both topological and feature spaces and constrains the diversity and consistency of node embedding by measuring the similarity between specific and common embedding. However, it is important to note that the aforementioned models cannot be directly applied to heterogeneous graphs.

HGNNs learn node representation by capturing information from different types of nodes and edges through metapaths or relation types. For instance, HAN (Heterogeneous graph attention network) [20] learns the importance between nodes and their neighbors under meta-paths through node-level attention and the importance of different meta-paths through semantic-level attention. Building upon HAN, MAGNN (Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding) [21] further enhances node representation by aggregating information from heterogeneous nodes within meta-paths. Additionally, HetGNN (Heterogeneous Graph Neural Network) [22] uses random walks with a restart to sample a fixed-size neighborhood and integrates features of the same or different types of nodes through a bidirectional LSTM (Long Short Term Memory). HGT (Heterogeneous Graph Transformer) [23] captures the importance of different types of edges by computing attention coefficients between nodes and aggregates edge attention with node information for message passing. Finally, HGSL (Heterogeneous Graph Structure Learning for Graph Neural Networks) [24] achieves heterogeneous graph structure learning by fusing multiple subgraphs (feature graph, semantic graph, and the original graph). ie-HGCN (Interpretable and Efficient Heterogeneous Graph Convolutional Network) [25] is a relation extraction model based on graph neural networks that uses a combination of various relation

representation methods, effectively capturing dependencies and contextual information between entities. RoHe (Robust Heterogeneous Graph Neural Networks against Adversarial Attacks) [26] employs an attention purifier that can prune malicious neighbors based on topology and features, thus eliminating the negative influence of malicious neighbors in the soft attention mechanism. HPN (Heterogeneous Graph Propagation Network) [27] is a graph neural network model for graph classification that enhances model performance through hierarchical graph pooling and structure learning, effectively handling graph structures at different levels.

2.2 Graph contrastive learning

Contrastive learning on graphs follows the principle of Mutual Information (MI) maximization [28], which aims to pull closer the representation of samples with similar information while pushing away the representation of unrelated samples [29]. In heterogeneous graph contrastive learning, it is common to perform MI maximization on samples at different scales (i.e., node-level and graph-level representation). HDGI (Heterogeneous Deep Graph Infomax) [30] fuses node representation under different meta-paths through semantic-level attention to form positive sample node representation and optimizes node representation by maximizing the mutual information between positive samples and graph-level representation. DMGI (Unsupervised Attributed Multiplex Network Embedding) [31] optimizes node representation by maximizing mutual information between subgraph-level representation learned under each relation subgraph and node-level representation. Additionally, a recent self-supervised heterogeneous graph neural network HeCo (Self-supervised Heterogeneous Graph Neural Network with Co-contrastive

Learning) [32] that maximizes node-level mutual information has attracted widespread attention. It employs collaborative contrastive learning from the perspectives of network schema and meta-paths to uncover more information in heterogeneous graphs. However, existing heterogeneous graph contrastive learning methods are only used in self-supervised models and cannot directly utilize label information.

Although these methods provide insightful solutions for utilizing labels, they still fail to capture the rich information contained in the labels.

This paper proposes a label-guided semi-supervised contrastive learning framework that integrates the rich label information into GNN learning by jointly learning the representation of nodes and labels.

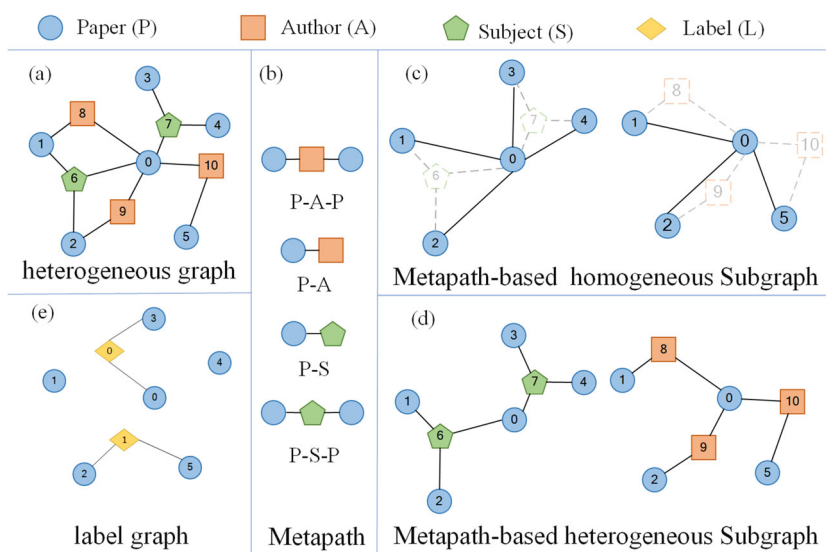
3 Preliminary

Definition 1. Heterogeneous Graph. A Heterogeneous graph is defined as $G = (V, E, \mathcal{A}, \mathcal{R}, \phi, \varphi)$. Where, V and E represent sets of nodes and edges, respectively. \mathcal{A} and \mathcal{R} represent sets of node types and edge types, where $|\mathcal{A} + \mathcal{R}| > 2$. There are two types of node mappings: $\phi : V \rightarrow \mathcal{A}$ for node types and $\varphi : E \rightarrow \mathcal{R}$ for edge types. V is defined as having two categories: labeled nodes V_L and unlabeled nodes V_U , where $V_L + V_U = V$. The labeled nodes are defined as V_L .

For example, Fig. 2(a) illustrates a heterogeneous graph composed of multiple types of nodes (paper, author, subject) and relationships (the writing relationship between author and paper, the purpose relationship between paper and subject).

Definition 2. Metapath. A metapath is defined as a path in a heterogeneous graph: $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$,

Fig. 2 An example of HIN



representing a composite connection relationship $R = R_1 \circ R_2 \circ \dots \circ R_l$ between A_1 and A_{l+1} , where \circ denotes the composition operator on relationships, $A_l \in \mathcal{A}$, $R_l \in \mathcal{R}$.

“path” typically refers to a sequence of connections between nodes in a graph, while “metapath” refers to a specific type of path pattern. For example, in Fig. 2(b), two metapaths PAP and PSP are illustrated. Where, PAP represents the connection where two papers share a common author, and PSP represents the connection where two papers jointly express the same topic.

Definition 3. Metapath-based Homogeneous Subgraph.

For a given heterogeneous graph G , a metapath P , and a node V , the homogeneous subgraph $G^{ho} = (V, E, \mathcal{A}, \mathcal{R}) \in G$ is defined as the graph constructed from all neighbor pairs based on metapath P . Please note that P starts and ends with the same node type, where $\mathcal{A} = 1$ and $\mathcal{R} = 1$.

For example, in Fig. 2(c), it shows the homogeneous subgraph generated by the two metapaths PAP and PSP, where the homogeneous subgraph only contains nodes of type paper.

Definition 4. Metapath-based Heterogeneous Subgraph.

Given a metapath P and a node V in a heterogeneous graph G , the metapath-based heterogeneous subgraph $G^{he} = (V, E, \mathcal{A}, \mathcal{R}) \in G$ is defined as the graph constructed by pairs of neighboring nodes of different types connected to node V through the metapath, where $|\mathcal{A} + \mathcal{R}| > 2$.

For example, in Fig. 2(d), it shows the heterogeneous subgraph based on two metapaths PAP and PSP. The heterogeneous subgraph under the metapath PAP contains only paper nodes and author nodes, while the heterogeneous subgraph under the metapath PSP contains only paper nodes and subject nodes.

Definition 5. Label Graph. The node label graph $G^Y \in \mathbb{R}^{M \times C}$ is composed of one-hot vectors for labeled nodes and zero vectors for unlabeled nodes, where M is the number of nodes in V , and C is the number of label classes. Specifically, each labeled node $V_i \in V_L$ has a one-hot vector $Y_i \in \{0, 1\}^C$, where 1 indicates the label category of V_i . For each unlabeled node $V_i \in V_U$, $Y_i \in \{0\}^C$ is a all-zero vector, where all elements are 0.

For example, in Fig. 2(e), the labeled node V_0 has $Y_0 = \{1, 0\}$, where 1 represents that the node V_0 belongs to label 0. In the label graph G^Y shown in Fig. 2(e), it is represented as

$$G^Y = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Definition 6. Node Embedding [21]. Node Embedding is a technique that maps nodes in a graph to a low-dimensional vector space, commonly used for representation learning of graph data.

For a node $v_i \in V$, its Node Embedding is denoted as $v_i = f(v_i, G)$, where f is a mapping function that maps node v_i and the entire graph G to a vector representation v_i in \mathbb{R}^d space. $v_i \in \mathbb{R}^d$, where d is the dimensionality of the chosen embedding space.

Problem. Heterogeneous Graph Embedding. Given a heterogeneous graph $G = (V, E, \mathcal{A}, \mathcal{R}, \phi, \varphi)$, with node attribute matrices X_{A_i} , heterogeneous graph embedding is the task to learn the d -dimensional node representations $Z \in \mathbb{R}^d$ with $d \ll |V|$ that are able to capture rich structural and semantic information involved in G .

4 The proposed method

In this section, we propose a Semi-Supervised Heterogeneous Graph Contrastive Learning with Label-Guided, as illustrated in Fig. 3. The model comprises three parts: (a) Metapath-based Heterogeneous Subgraph, (b) Metapath-based Homogeneous Subgraph, and (c) contrastive learning. Specifically, to better capture information from homogeneous and heterogeneous neighbors in the heterogeneous graph, SSGCL-LG decomposes the graph into multiple metapath-based homogeneous and heterogeneous subgraphs. In (a) Metapath-based Heterogeneous Subgraph, information from different metapath-based heterogeneous subgraphs is aggregated using attention mechanisms. In (b) Metapath-based Homogeneous Subgraph, a label graph is constructed and concatenated with the homogeneous subgraphs to learn node representations using a GNN Encoder. Finally, in (c) contrastive learning, different positive and negative samples are selected, optimizing the model through a combination of contrastive loss and cross-entropy loss.

4.1 Metapath-based heterogeneous subgraph embedding

Most current research on heterogeneous graphs is based on metapaths, used to capture specific semantic information in graphs. However, these heterogeneous graph models are primarily constrained by two limitations: firstly, many of them only aggregate information from homogeneous neighbors connected by meta-paths, thereby discarding rich structural and attribute information from heterogeneous neighbors; secondly, some studies aggregate information from both homogeneous and heterogeneous neighbors but treat these neighbors indiscriminately in the same way. As a result, these methods may lose important information and lead to sub-

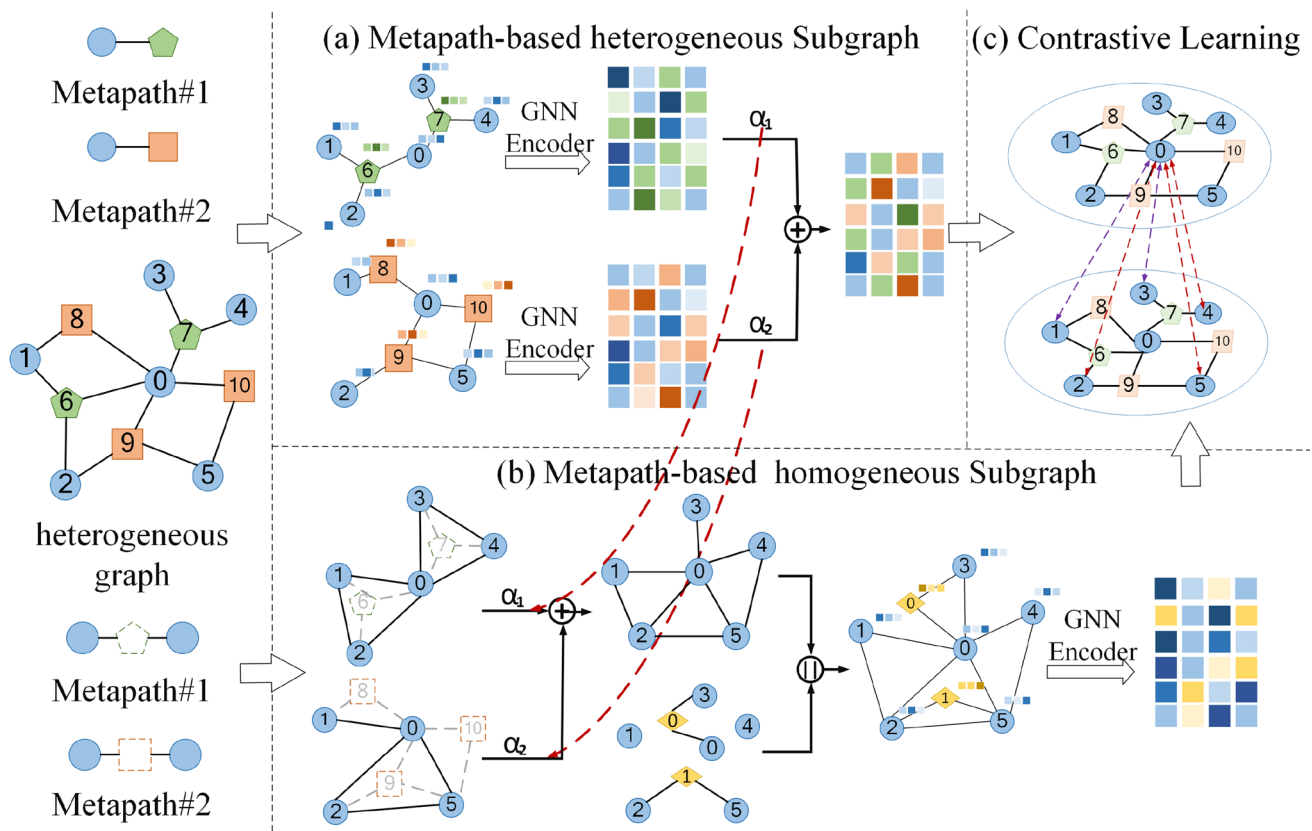


Fig. 3 SSGCL-LG model

optimal performance. We illustrate this point through the following example. Therefore, the heterogeneous graph is first partitioned into homogenous and heterogeneous subgraphs based on Metapath, enabling the comprehensive learning of complex information in the heterogeneous graph.

As shown in Fig. 2(c) of the heterogeneous graph constructed with the metapath PAP, a homogeneous subgraph is formed with paper nodes (V_0, V_1, V_2, V_5). For a specific node (e.g., V_0), if we only aggregate information from homogeneous neighbors (V_1, V_2, V_5), the structural and attribute information contributed by heterogeneous neighbors (V_8, V_9, V_{10}) connected to it will be ignored. Therefore, considering only homogeneous subgraphs can lead to the loss of a significant amount of useful interaction information from the original graph. Additionally, there are different interaction patterns between nodes and neighbors of different types, which often carry different semantics and should be considered separately to avoid information loss. It's worth noting that nodes of different types typically have different attributes. For example, in a recommendation system, user node attributes may include age, gender, interests, while item attributes may include price, text descriptions, images, etc. Original attributes cannot be directly transferred between nodes of different types and require pre-transformation.

Since nodes of different types in HIN usually have different vector dimensions, in SSGCL-LG, they need to be projected into a common space through specific transformation. Additionally, we treat labels as a special type of node and initializes the feature of label nodes with unit vectors. Specifically, for nodes of type Φ , a specific type of mapping matrix w_Φ is designed to transform their features X into the common space, as shown below:

$$H = \sigma(w_\Phi X + b_\Phi), \tag{1}$$

where σ represents the activation function, and b denotes the vector bias.

Different types of metapaths represent different semantic information. For M types of metapaths, SSGCL-LG constructs heterogeneous subgraphs of this type $\{G_1^{he}, \dots, G_M^{he}\}$. For each subgraph G_n^{he} , node embedding H_n^{he} are learned using GCN [11]. Specifically:

$$\left(H_n^{he}\right)^{l_{he}} = \sigma\left(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}}\left(H_n^{(l-1)}\right) W^{(l)}\right), \tag{2}$$

where $\hat{A} = A + I$ represents the adjacency matrix of the heterogeneous subgraphs G_n^{he} with the addition of self-loop

connections. \hat{D} is the degree matrix of \hat{A} . $W^{(l)}$ denotes the weight matrix for the l -th layer of the Graph Convolutional Network, and $(H_n^{he})^{l_{he}}$ represents the node representation at the l -th layer.

After obtaining embedding for each type of heterogeneous subgraphs $\{h_1^{he}, \dots, h_M^{he}\}$, SSGCL-LG utilizes semantic-level attention to fuse them, resulting in node Z^{he} under the heterogeneous subgraph:

$$Z^{he} = \sum_{n=1}^M \alpha_n \cdot h_n^{he}, \quad (3)$$

where α_n represents the weight of the heterogeneous subgraphs G_n^{he} , calculated as follows:

$$e_{G_n^{he}} = \frac{1}{|M|} \sum_{n \in M} a^T \tanh(W \cdot H_n^{he} + b), \quad (4)$$

$$\alpha_n = \text{softmax}\left(e_{G_n^{he}}\right) = \frac{\exp\left(e_{G_n^{he}}\right)}{\sum_{n \in M} \exp\left(e_{G_n^{he}}\right)}. \quad (5)$$

Note that, because the importance of heterogeneous subgraphs varies across different metapaths, we can compute weights for aggregating different heterogeneous subgraphs, denoted as $\{\alpha_1, \dots, \alpha_M\}$.

4.2 Metapath-based homogeneous subgraph embedding

For different types of metapaths representing distinct semantic information, SSGCL-LG constructs homogeneous subgraphs $\{G_1^{ho}, \dots, G_M^{ho}\}$. According to the number of metapaths, weight matrices for constructing homogeneous subgraphs $\{w_1^{ho}, \dots, w_M^{ho}\}$ are formed.

When integrating different views traditionally, the channel attention method described in HGSL [24] is employed:

$$w_{ho} = \Psi[w_1^{ho}, \dots, w_M^{ho}], \quad (6)$$

where Ψ represents a channel attention layer with parameters $W^\Psi \in \mathbb{R}^{1 \times 1 \times M}$. It performs a 1×1 convolution on the input using $\text{softmax}(W^\Psi)$.

However, this method only utilizes softmax for the 1×1 convolution operation, neglecting the influence of node features on graph structure fusion. To account for the impact of node features, we utilize semantic-level attention within heterogeneous subgraphs, resulting in different weight coefficients $\{\alpha_1, \dots, \alpha_M\}$ guiding the construction of the weight

matrix w_{ho} for homogeneous subgraphs,

$$w^{ho} = \sum_{n=1}^M \alpha_n w_n^{ho}. \quad (7)$$

SSGCL-LG achieves this by creating a type of node for the labels and establishing connections with nodes within the same class. It constructs a connection matrix G^Y between labels and nodes within the same class. G^Y is concatenated with w^{ho} to create the label heterogeneous adjacency matrix w^{la} . Similarly, GCN [11] is utilized to learn node embedding representation Z^{ho} within the label subgraph,

$$w^{la} = \begin{bmatrix} w^{ho} & G^Y \\ G^{YT} & 0 \end{bmatrix}, \quad (8)$$

$$Z^{ho} = \sigma(\hat{D}^{-\frac{1}{2}} w^{la} \hat{D}^{-\frac{1}{2}} H^{(l_{ho}-1)} W^{(l_{ho})}). \quad (9)$$

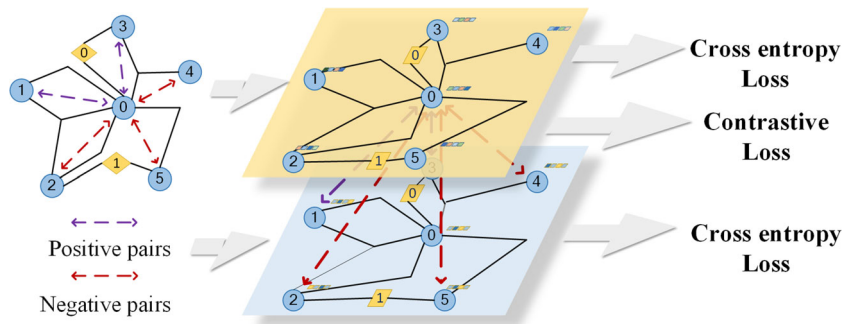
In this work, we use one-hot encoding to represent label features, which can provide rich representations when label features or prior knowledge related to labels are explicitly given. When further performing message passing on w^{la} , labels can contribute in two aspects. Firstly, each label serves as a virtual center for intra-class nodes, making them 2-hop neighbors even if they are far apart from each other in w^{la} . This enhances the smoothness of intra-class node representations. Secondly, modeling label semantics through one-hot encoding helps in discovering semantic correlations among labels. Although there are no direct connections between labels, they can still receive messages from each other through higher-order interactions, aiding in uncovering their implicit relationships.

4.3 Positive sample selection strategy

In the selection of positive samples, as illustrated in Fig. 4, considering that only a small number of nodes have label information, nodes with the same label are considered positive samples, and nodes with different labels are considered negative samples. Additionally, since nodes are typically connected by multiple paths and are highly correlated, we propose a positive selection strategy: if there are multiple metapaths connecting two nodes, they are considered positive samples. This is depicted in Fig. 4, where links between papers indicate that they are positive samples for each other. One advantage of this strategy is that the selected positive samples can better reflect the local structure of the target node.

In Fig. 4, for example, node V_0 and node V_3 belong to label 0, while node V_2 and node V_5 belong to label 1. Thus, node V_0 and node V_3 are positive samples for each other, and

Fig. 4 Positive sample selection strategy



they are negative samples for node V_2 and node V_5 . Among other nodes with unknown labels, node V_0 has two paths connecting to node V_1 and one path connecting to node V_4 . Assuming the threshold T is set to 2, then node V_0 and node V_1 are positive samples for each other, and node V_0 and node V_4 are negative samples for each other. Thus, the positive sample set for node V_0 is $[V_1, V_3]$, and the negative sample set is $[V_2, V_4, V_5]$.

If there is a metapath connecting two nodes, then these two nodes are related. The more metapaths between two nodes, the stronger their correlation. For nodes i and j , we define a function $C_i(\cdot)$ to count the number of metapaths connecting these two nodes:

$$C_i(j) = \sum_{n=1}^M \theta(j \in N_i), \tag{10}$$

where $\theta(\cdot)$ denotes the sign function. We construct a set C_i sorted in descending order according to $C_i = \{j \mid j \in V \text{ and } C_i(j) \neq 0\}$ nodes as candidate positive samples \mathbb{P}_i^T .

To make full use of the limited label information available for some nodes, we consider nodes with the same label as positive samples and nodes with different labels as negative samples. Specifically, we construct a set $\mathbb{Q}_i = \{j \mid j \in V \text{ and } Q_i(j) = 1\}$, where $Q_i(j) \in \{0, 1\}$ represents the label discrimination function (when nodes i and j have the same label, $Q_i(j) = 1$, otherwise $Q_i(j) = 0$). The final positive samples are filtered based on both metapaths and label information, denoted as $\mathbb{P}_i = \mathbb{P}_i^T \parallel \mathbb{Q}_i$, and the remaining nodes serve as negative samples \mathbb{N}_i .

4.4 Training

The semi-supervised contrastive loss is an extension of the self-supervised contrastive loss. As evident from the selection of positive samples, the incorporation of label information expands the number of positive node pairs in semi-supervised contrastive learning.

Once obtaining the embedding z_i^{ho} for the homogeneous subgraphs and z_i^{he} for the heterogeneous subgraphs, we feed

them into a MLP with one hidden layer to map them into the space where contrastive loss is calculated:

$$z_i^{ho\ proj} = w^{(2)} \sigma \left(w^{(1)} z_i^{ho} + b^{(1)} \right) + b^{(2)}, \tag{11}$$

$$z_i^{he\ proj} = w^{(2)} \sigma \left(w^{(1)} z_i^{he} + b^{(1)} \right) + b^{(2)}, \tag{12}$$

where σ is the activation function, $w^{(1)}$ is the weight matrix for the first layer, used to map input $z^{(i)}$ to the output of the hidden layer, $b^{(1)}$ is the bias term for the first layer, used to adjust the influence of the input, $w^{(2)}$ is the weight matrix for the second layer, used to map the output of the first layer to the output layer, $b^{(2)}$ is the bias term for the second layer, used to adjust the influence of the output of the first layer.

After obtaining the positive sample set \mathbb{P}_i and negative sample set \mathbb{N}_i , the loss for the homogeneous subgraphs is computed as:

$$\mathcal{L}_i^{ho} = - \log \frac{\sum_{j \in \mathbb{P}_i} \exp \left(\text{sim} \left(z_i^{ho\ proj}, z_j^{he\ proj} \right) / \tau \right)}{\sum_{k \in \{\mathbb{P}_i \cup \mathbb{N}_i\}} \exp \left(\text{sim} \left(z_i^{ho\ proj}, z_k^{he\ proj} \right) / \tau \right)}, \tag{13}$$

where $\text{sim}(u, v)$ is the cosine similarity function, and τ is an environmental variable.

In the homogeneous subgraphs perspective, the target embedding $z_i^{ho\ proj}$ comes from the homogeneous subgraphs perspective, while the positive and negative sample embedding $z_k^{he\ proj}$ come from the heterogeneous subgraphs perspective.

Similarly, the loss in the heterogeneous subgraphs perspective is:

$$\mathcal{L}_i^{he} = - \log \frac{\sum_{j \in \mathbb{P}_i} \exp \left(\text{sim} \left(z_i^{he\ proj}, z_j^{ho\ proj} \right) / \tau \right)}{\sum_{k \in \{\mathbb{P}_i \cup \mathbb{N}_i\}} \exp \left(\text{sim} \left(z_i^{he\ proj}, z_k^{ho\ proj} \right) / \tau \right)}. \tag{14}$$

The difference lies in the fact that the target embedding $z_i^{he\ proj}$ comes from the heterogeneous subgraphs perspective, while the positive and negative sample embedding

$z_k^{ho} proj$ come from the homogeneous subgraphs perspective.

Therefore, the contrastive loss is as follows:

$$L_{con} = \frac{1}{|V|} \sum_{i \in V} [\lambda \cdot \mathcal{L}_i^{ho} + (1 - \lambda) \cdot \mathcal{L}_i^{he}], \tag{15}$$

where λ is used to balance the losses from the two subgraphs.

The cross-entropy loss function can be described as:

$$L_{he} = - \sum_{v \in y_l} Y_{v_l} \cdot \ln(C \cdot Z^{he}), \tag{16}$$

$$L_{ho} = - \sum_{v \in y_l} Y_{v_l} \cdot \ln(C \cdot Z^{ho}), \tag{17}$$

where y_L represents the labeled nodes set, Y_{v_l} is the true label of node v_l , and C represents the parameters of the classifier.

The cross-entropy loss is defined as follows:

$$L_{cro} = uL_{he} + (1 - \mu)L_{ho}, \tag{18}$$

where u is used to balance the losses from two subgraphs.

Finally, by combining the contrastive loss L_{con} and the cross-entropy loss L_{cro} , our overall loss function for SSGCL-LG can be represented as follows:

$$L = \tau L_{con} + (1 - \tau)L_{cro}. \tag{19}$$

Algorithm 1 describes the main flow of the enhanced representation of the target node attributes.

5 Experiments

In this section, we conduct extensive experiments to demonstrate the performance of SSGCL-LG. Specifically, we show the excellent performance of our method through node classification, node clustering, and visualization. Additionally, label importance analysis experiments, ablation experiments, and parameter analysis experiments further prove the effectiveness of SSGCL-LG.

5.1 Datasets

To evaluate the effectiveness of the proposed framework for attribute completion, we utilize three common Heterogeneous Information Network (HIN) datasets. Table 1 summarizes the statistical data of these three datasets.

Algorithm 1 The algorithm of attribute enhancement.

Input: the heterogeneous graph G , the node feature X , the heterogeneous Subgraphs G^{he} , the heterogeneous Subgraphs G^{ho} , the label graph G^Y

Output: the node embedding Z^{he} and Z^{ho}

- 1: Project the node feature X into a unified dimension according to (1).
- 2: **for** $G_n^{he} \in \{G_1^{he}, \dots, G_M^{he}\}$ **do**
- 3: Calculate the heterogeneous Subgraphs embedding H_n^{he} according to (4);
- 4: **end for**
- 5: **for** $n \in M$ **do**
- 6: Calculate the weight of heterogeneous Subgraphs α_n according to (6)
- 7: **end for**
- 8: Calculate different heterogeneous Subgraphs embedding Z^{he} according to metapaths
- 9: Z^{he} and $\alpha_n \in \{\alpha_1, \dots, \alpha_M\}$
- 10: **for** $G_n^{ho} \in \{G_1^{ho}, \dots, G_M^{ho}\}$ **do**
- 11: Calculate heterogeneous Subgraphs Weight matrix according to the number of metapaths w_n^{ho} ;
- 12: **end for**
- 13: Calculate homogeneous Subgraphs w^{ho} according to (7).
- 14: Connect homogeneous Subgraphs w^{ho} and label graph G^Y according to (8);
- 15: Calculate the heterogeneous Subgraphs embedding H_n^{he} according to (9).
- 16: Return Z^{ho}

Table 1 Statistics of datasets

Datasets	Nodes	Edges	Metapaths
ACM	Paper(P)	P-A:13407	PAP
	Author(A)	P-S:4019	PSP
	Subject(S)		
IMDB	Movie(M)	M-D:4278	MAM
	Director(D)	M-A:12828	MDM
	Actor(A)		
DBLP	Author(A)	P-A:19645	APA
	Paper(P)	P-V:14328	APVPA
	Venue(V)	P-T:85810	APTPA
	Term(T)		

- **ACM¹**: This is an academic network that includes three different types of nodes: 4,019 papers, 7,167 authors, and 60 subjects. The target nodes are papers, which are categorized into three different classes.
- **IMDB²**: This is a movie network that comprises three different types of nodes: 4,278 movies, 2,081 directors, and 5,257 actors. The target nodes are movies, which are categorized into three different classes

¹ <https://dl.acm.org/>

² <http://www.imdb.com/>

- **DBLP³**: This is also an academic network, containing four different types of nodes: 4,057 authors, 14,328 papers, 20 conferences, and 7,723 terms. The target nodes are authors, which are categorized into four different classes.

5.2 Baselines

We compare the proposed SSGCL-LG with three categories of baselines: Method based on homogeneous graph (GCN [11], GAT [13]), Method based on metapaths (HAN [20], RoHe [26], MAGNN [21], HPN [27]), Method based on Relation-aware (HGSL [24], HGT [23], ie-HGCN [25]).

- **GCN(2017)** [11]: A semi-supervised graph convolutional network primarily designed for homogeneous graphs. In this paper, GCN is applied to all meta-paths of heterogeneous graphs and achieves the best performance.
- **GAT(2018)** [13]: It employs a multi-head attention mechanism to assign weights to each neighboring node, mainly targeting homogeneous graphs. In this paper, GAT is applied to all meta-paths of heterogeneous graphs and achieves the best performance.
- **HAN(2019)** [20]: This model generates node embedding by performing hierarchical aggregation of neighborhood features based on meta-paths, learning the importance from both the node level and the semantic level.
- **MAGNN(2020)** [21]: This model generates node embedding by applying node content transformation, intra-meta-path aggregation, and inter-meta-path aggregation.
- **HGT(2020)** [23]: It introduces an attention mechanism related to vertex and edge types.
- **ie-HGCN(2023)** [25]: ie-HGCN is a relation extraction model based on graph neural networks that uses a combination of various relation representation methods, effectively capturing dependencies and contextual information between entities.
- **HGSL(2021)** [24]: It generates a heterogeneous graph structure suitable for downstream tasks by mining feature similarity, the interaction between features and structure, and the high-order semantic structure in heterogeneous graphs, and jointly learns GNN parameters.
- **RoHe(2022)** [26]: RoHe employs an attention purifier that can prune malicious neighbors based on topology and features, thus eliminating the negative influence of malicious neighbors in the soft attention mechanism.
- **HPN(2022)** [27]: HPN is a graph neural network model for graph classification that enhances model performance through hierarchical graph pooling and structure learning, effectively handling graph structures at different levels.

- **SSGCL-LG(ours)**: It integrates label information into the learning process of graph neural networks by constructing a labeled graph and building positive samples related to labels.

5.3 Metrics

In this study, we employed multiple evaluation metrics to assess the performance of the models. These metrics cover different aspects of model performance, including classification accuracy, clustering consistency, and class distribution.

- **Micro-F1**: Micro-F1 is one of the commonly used evaluation metrics in multi-class classification tasks. It combines precision and recall and is suitable for datasets with imbalanced class distributions. The formula for Micro-F1 is as follows:

$$\text{Micro-F1} = \frac{2 \times (\text{Micro-Precision} \times \text{Micro-Recall})}{\text{Micro-Precision} + \text{Micro-Recall}}$$

Where, Micro-Precision represents micro-precision, defined as the ratio of correct predictions for all classes to all predicted instances. Micro-Recall represents micro-recall, defined as the ratio of correct predictions for all classes to all true labels.

- **Macro-F1**: Macro-F1 is another commonly used evaluation metric in multi-class classification tasks, which computes the average F1 score for each class. The formula for Macro-F1 is as follows:

$$\text{Macro-F1} = \frac{1}{N} \sum_{i=1}^N \frac{2 \times (\text{Precision}_i \times \text{Recall}_i)}{\text{Precision}_i + \text{Recall}_i}$$

Where, N denotes the number of classes, Precision_i and Recall_i represent precision and recall, respectively, for class i .

- **NMI (Normalized Mutual Information)**: NMI is a commonly used evaluation metric in clustering tasks, measuring the consistency between clustering results and true labels. The formula for NMI is as follows:

$$\text{NMI} = \frac{I(X; Y)}{\sqrt{H(X) \times H(Y)}}$$

Where, $I(X; Y)$ denotes mutual information, measuring the correlation between two random variables X and Y ; $H(X)$ and $H(Y)$ denote the entropy of random variables X and Y , respectively.

- **ARI (Adjusted Rand Index)**: ARI is another commonly used evaluation metric in clustering tasks, evaluating clustering effectiveness by comparing the consistency

³ <https://github.com/cynricfu/MAGNN>

between clustering results and true labels with the consistency between random clustering results and true labels. The formula for ARI is as follows:

$$ARI = \frac{\sum_{i,j} \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{n}{2}}{\frac{1}{2} [\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{n}{2}}$$

Where, n_{ij} represents the number of samples simultaneously belonging to class i and class j ; a_i represents the number of samples belonging to class i in clustering results; b_j represents the number of samples belonging to class j in true labels; n represents the total number of samples.

These evaluation metrics comprehensively consider the model's performance in classification and clustering tasks, providing important references for the objective assessment of research results.

5.4 Experimental setting

To ensure fairness, we use the same training, validation, and testing sets for all methods in this study. Moreover, we set the same dimensional embedding for all methods compared. The hidden layer dimension is set to 64 for all compared methods. The attention mechanism is extended to multi-head attention, and the number of attention heads K is set to 8, as this is found experimentally to produce more stable results.

Node classification experiments, node clustering experiments, ablation experiments, attention analysis experiments, and parameter analysis experiments all utilized the ACM, IMDB, and DBLP datasets. Label importance analysis used the IMDB dataset.

5.5 Node classification

In this section, we first evaluate the node classification results of SSGCL-LG in a semi-supervised setting. Specifically, we input the node representation into a Support Vector Machine (SVM) for classification, dividing the data into different training ratios from 20% to 80%, and using Micro-F1 and Macro-F1 as evaluation metrics. Conduct five repeated experiments and report the average results. The best results are highlighted in bold. The results are shown in Table 2.

Models based on heterogeneous graphs typically outperform models based on homogeneous graphs (GCN, GAT). It is evident that directly applying homogeneous graph models to heterogeneous graphs is not feasible, as heterogeneous graphs contain a greater variety of node and edge types, more complex information, necessitating research into more suitable heterogeneous graph models.

Compared to metapath-based heterogeneous graph models (HAN, RoHe, MAGNN, HPN), on the ACM dataset,

Macro-F1 and Micro-F1 have increased by 1.5% and 1.4% respectively compared to HAN, by 1.4% and 1% respectively compared to RoHe, by 0.8% and 1% respectively compared to MAGNN, and by 1% and 1.7% respectively compared to HPN. On the IMDB dataset, Macro-F1 and Micro-F1 have increased by 3.4% and 3.0% respectively compared to HAN, by 3.0% and 2.2% respectively compared to RoHe, by 2.2% and 2.2% respectively compared to MAGNN, and by 1.7% and 1.7% respectively compared to HPN. On the DBLP dataset, Macro-F1 and Micro-F1 have increased by 2.2% and 2.8% respectively compared to HAN, by 2.8% and 2.2% respectively compared to RoHe, by 1.4% and 1.4% respectively compared to MAGNN, and by 2.2% and 2.2% respectively compared to HPN. The reason for these improvements is that HAN, RoHe, MAGNN and HPN are models built on homogeneous graphs derived from meta-paths, considering only the information of the target nodes and ignoring the information of other types of nodes. SSGCL-LG, on the other hand, decomposes the heterogeneous graph into multiple meta-path-based subgraphs of both homogeneous and heterogeneous types, which allows it to better capture the information of both homogeneous and heterogeneous neighbors in the heterogeneous graph.

Compared to metapath-based heterogeneous graph models (HGSL, HGT, ie-HGCN), on the ACM dataset, Macro-F1 and Micro-F1 have increased by 0.7% and 1.7% respectively compared to HGSL, by 1.7% and 1.7% respectively compared to HGT, and by 0.7% and 0.7% respectively compared to ie-HGCN. On the IMDB dataset, Macro-F1 and Micro-F1 have increased by 3.1% and 4.2% respectively compared to HGSL, by 4.2% and 4.2% respectively compared to HGT, and by 0.7% and 0.7% respectively compared to ie-HGCN. On the DBLP dataset, Macro-F1 and Micro-F1 have increased by 1.3% and 6% respectively compared to HGSL, by 6% and 6% respectively compared to HGT, and by 1.2% and 1.2% respectively compared to ie-HGCN.. The reason for these improvements is that while HGSL, HGT and ie-HGCN although consider the information of heterogeneous nodes, they only use labels for calculating loss, and the learning process cannot access label information. SSGCL-LG, on the other hand, encodes labels into the learning process of the graph neural network, fully considering the information of the labels.

5.6 Node clustering

In this section, the K-means method is employed to cluster the embedding vectors obtained from the model. The parameter K for K-means is set to the number of label categories in the dataset, which corresponds to the actual number of node categories. The clustering results are evaluated using NMI (Normalized Mutual Information) and ARI (Adjusted Rand Index). NMI measures the closeness of the clustering results,

Table 2 Experiment results (%) for the node classification task

Datasets	Metrics	Ratio	GCN	GAT	HAN	MAGNN	HGT	HGSL	RoHe	HPN	ie-HGCN	SSGCL-LG
ACM	Macro-F1	20%	90.51	89.09	90.71	90.02	91.53	92.43	91.22	92.08	91.35	92.65
		40%	90.68	89.32	91.33	91.39	91.68	92.58	91.61	92.27	92.14	93.22
		60%	90.8	89.43	91.73	92.18	91.81	92.73	92.09	92.52	92.59	93.52
		80%	90.58	89.33	91.91	92.67	91.82	92.83	92.08	92.52	92.79	93.51
	Micro-F1	20%	90.49	89.2	90.59	89.94	91.62	92.38	91.13	91.96	91.27	92.73
		40%	90.68	89.4	91.22	91.38	91.79	92.54	91.52	92.16	92.11	93.13
		60%	90.79	89.49	91.6	92.13	91.91	92.69	91.97	92.39	92.53	93.34
		80%	90.56	89.4	91.76	92.61	91.9	92.77	91.94	92.38	92.73	93.39
IMDB	Macro-F1	20%	49.03	58.6	58.11	57.87	57.14	58.14	58.05	60.72	58.24	61.49
		40%	49.15	58.67	58.56	59.23	57.38	58.07	58.61	61.25	59.33	62.01
		60%	49.71	58.78	58.73	59.72	57.63	58.51	59.07	61.42	59.65	62.24
		80%	49.94	58.6	58.88	59.94	57.99	59.13	59.01	61.57	59.87	62.24
	Micro-F1	20%	49.43	58.74	58.14	57.89	57.4	58.26	58.3	60.67	58.16	61.53
		40%	49.63	58.84	58.58	59.29	57.6	58.05	58.88	61.22	59.26	62.03
		60%	49.95	58.92	58.72	59.8	57.78	58.47	59.33	61.38	59.57	62.24
		80%	50.12	58.8	58.91	60.06	57.72	59.09	59.29	61.58	59.82	62.27
DBLP	Macro-F1	20%	89.04	89.76	92.63	93.75	88.65	93.72	91.94	92.83	92.73	94.7
		40%	89.05	89.75	92.87	93.83	88.98	93.65	92.22	92.88	93.57	95.05
		60%	89.01	89.77	93.05	93.81	89.22	93.81	92.29	92.99	93.66	95.05
		80%	89.17	89.83	93.16	94.1	89.37	94.09	92.61	93.16	94.09	95.37
	Micro-F1	20%	89.71	90.53	93.2	94.2	89.7	94.19	92.44	93.38	93.24	95.08
		40%	89.72	90.53	93.43	94.26	89.99	94.09	92.73	93.43	94	95.43
		60%	89.7	90.56	93.61	94.25	90.25	94.23	92.83	93.55	94.1	95.44
		80%	89.95	90.61	93.69	94.51	90.4	94.52	93.11	93.69	94.47	95.72

while ARI reflects the degree of overlap in the partitioning. The closer the NMI or ARI results are to 1, the better the clustering results are considered to be. The experimental results are shown in Table 3, with the optimal results highlighted in bold.

From the Table 3, it is evident that the SSGCL-LG model generally outperforms other models. Analysis indicates that our model fully takes into account the information of node

labels and, through contrastive learning, regards nodes with the same label as positive samples. This approach allows nodes of the same category to cluster more effectively, hence demonstrating better clustering performance.

To perform the visualization task and provide a more intuitive comparison, we learn the node embedding of the aforementioned methods (i.e., MAGNN, ie-HGCN, HGSL, HPN, RoHe, SSGCL-LG) on the DBLP dataset and project

Table 3 Experiment results (%) for the node clustering task

Datasets	ACM		DBLP		IMDB	
Metrics	NMI	ARI	NMI	ARI	NMI	ARI
HAN	0.7125	0.7515	0.7278	0.7833	0.1196	0.1197
MAGNN	0.7016	0.7214	0.7867	0.8400	0.1308	0.1276
HGT	0.6409	0.6646	0.6628	0.6543	0.1446	0.1176
HGSL	0.7025	0.7425	0.7632	0.7938	0.0621	0.0878
RoHe	0.6492	0.6810	0.6290	0.6984	0.1239	0.1294
HPN	0.7185	0.7598	0.7906	0.8479	0.1563	0.1498
ie-HGCN	0.4947	0.3489	0.3233	0.2721	0.1308	0.1304
SSGCL-LG	0.7357	0.7562	0.8211	0.8822	0.1658	0.1712

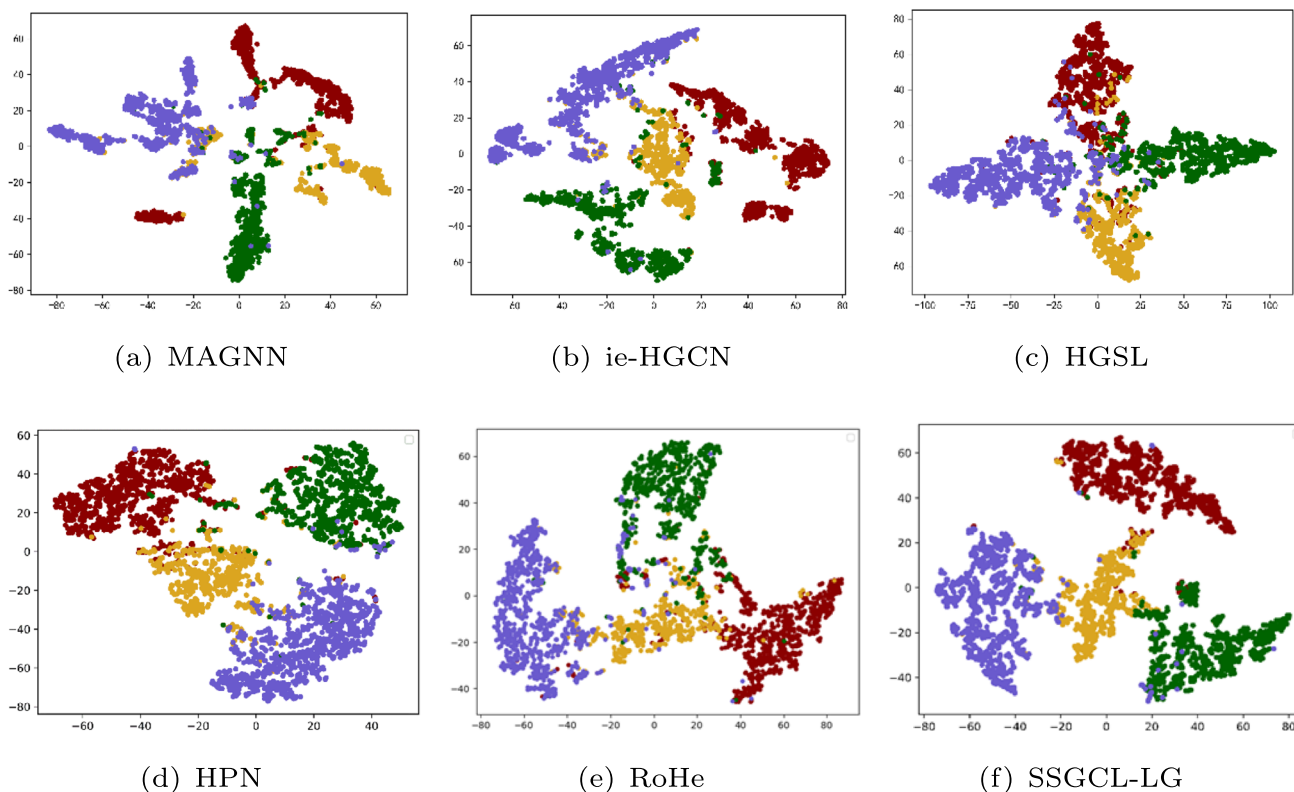


Fig. 5 Visualization

the embedding into two-dimensional space. Then, we used T-SNE to visualize the paper embedding in DBLP, coloring the nodes based on their classes.

As shown in Fig. 5, SSGCL-LG demonstrates clearer boundaries and denser clustering structures, which helps to distinguish different categories in visualization. This indicates that labels can contain rich information. By integrating label information into the learning process of node representation through the label graph, we can effectively distinguish nodes of different categories, significantly improving the model’s performance, and effectively differentiating papers belonging to different research fields.

5.7 Ablation experiments

To verify the effectiveness of different components of SSGCL-LG, we designed three variants of SSGCL-LG and compared their classification performance with SSGCL-LG.

The notation is shown in Table 4, and the comparison results are shown in Fig. 6.

From Fig. 6, it can be seen that the performance of the complete SSGCL-LG model is superior to that of its variants. The SSGCL-LG model integrates label information into the learning process of the neural network by constructing a label graph. Indeed, labels contain valuable information that is beneficial for node classification. Additionally, during the contrastive learning process, SSGCL-LG treats nodes with the same label as positive samples for each other, aiming to utilize the supervisory information present in the existing data for network training. By leveraging the supervisory signals contained in both labeled and unlabeled data, the SSGCL-LG model can learn node representation more effectively. This learning approach ensures that nodes from the same class are more closely clustered together in the representation space, making them more distinguishable from nodes of different classes.

Table 4 Description of the ablation experiments symbol

-w/o-w label	Delete the label in the label graph
-w/o-pos label	Delete the label in the positive sample
-w/o-w label-pos label	Delete the labels in the label diagram and the positive sample

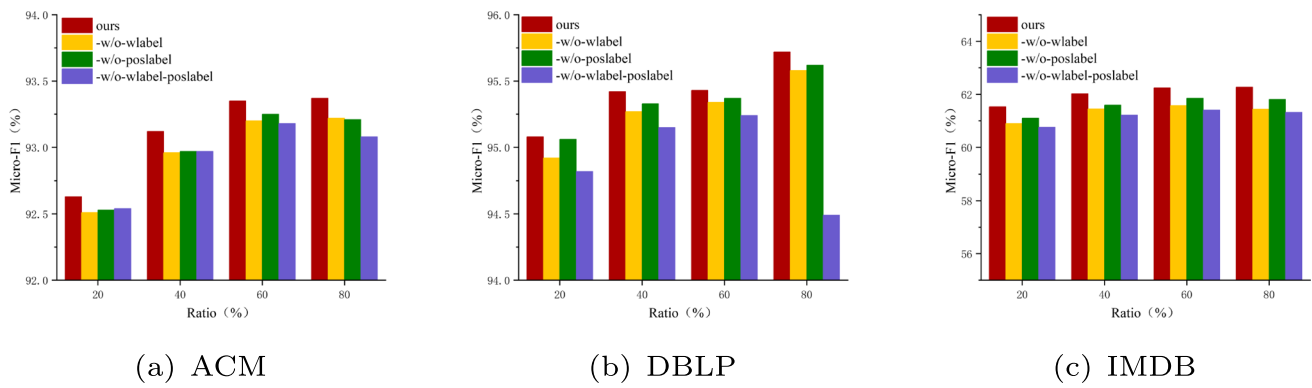


Fig. 6 Ablation experiments

5.8 Label importance analysis

To verify the importance of labels in the model, we re-divided the training set and selected subsets of different proportions for experiments on the IMDB dataset. The experimental results, as shown in Fig. 7(a), indicate that as the number of training samples increases, the performance of the model also gradually improves. This demonstrates that the quantity of labels has a significant impact on the performance of the model. Notably, among all the models compared, our model exhibits the most outstanding performance.

To further verify the effectiveness of labels, we conduct an ablation experiment by removing the labels from the model, including the labels in the heterogeneous node graph and the positive samples. The results, as shown in Fig. 7(b), indicate that when the number of training samples is very small, the performance of the ablated model is superior to that of the complete model. Our analysis find that when the number of training samples is very small, the label graph is too sparse, causing the model to fail to learn effective information from

the labels, thereby reducing the model's performance; as the number of training samples increases, the model can gradually learn more label information, hence the performance also gradually improves.

Compared to the ablated model, the performance improvement of our model is more pronounced. Because, as the number of training samples increases, the number of available labels also increases, allowing the model to learn useful information from the labels more effectively.

5.9 Attention analysis

To verify the effectiveness of the strategy where heterogeneous subgraphs under the meta-path guide the fusion of homogeneous subgraphs during aggregation, as opposed to aggregating under the meta-path's homogeneous subgraphs alone, we compared the guided fusion strategy (ours) with a channel attention strategy (ours-channel attention). The results of the comparison are shown in the Fig. 8.

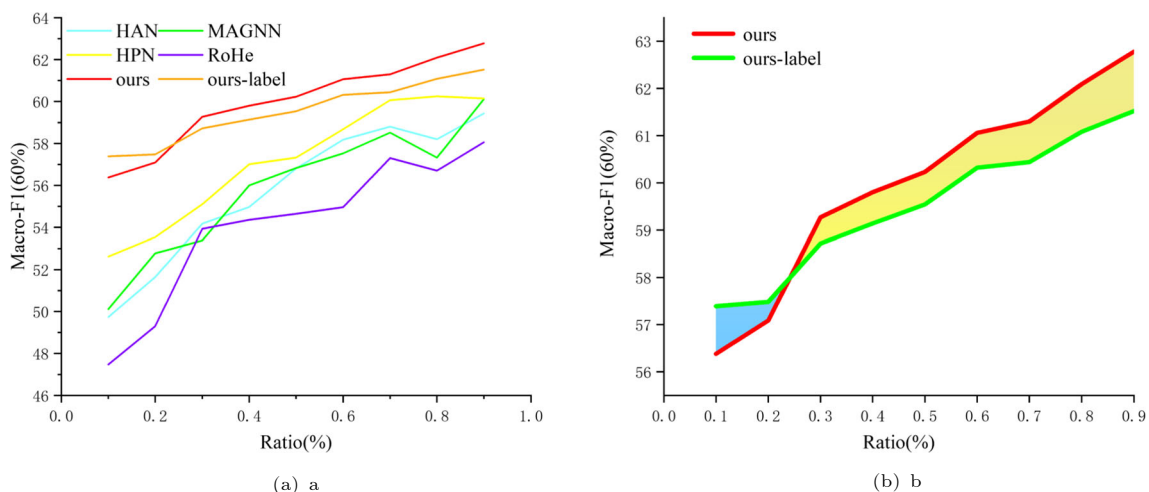


Fig. 7 Label importance analysis

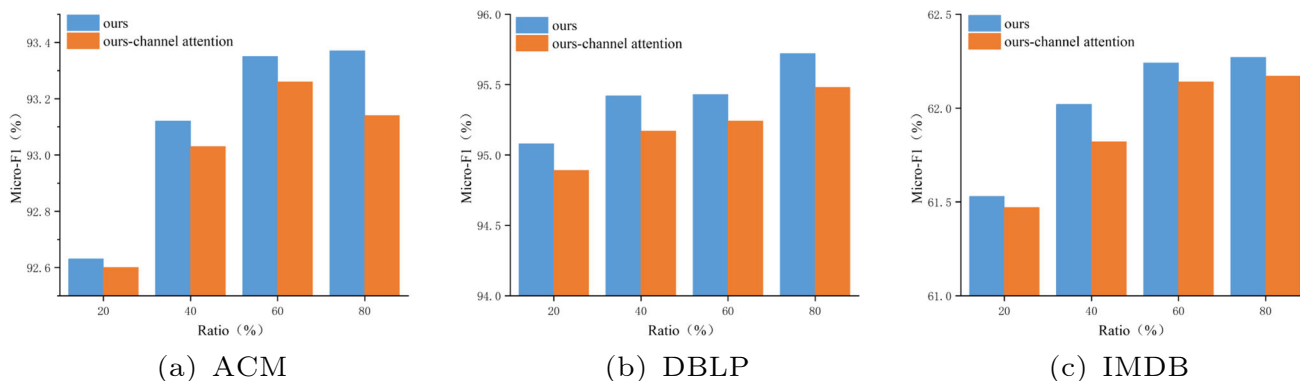


Fig. 8 Attention analysis

From Fig. 8, it can be seen that the strategy of using heterogeneous graphs to guide the fusion of homogeneous subgraphs is more effective than using channel attention.

In heterogeneous graphs, heterogeneous subgraphs are composed of nodes with specific meta-paths. Heterogeneous subgraphs can provide richer local structural information because they include interactions between all types of nodes within the meta-path. During the learning process of heterogeneous subgraphs, semantic-level attention is used to fuse representation under different heterogeneous subgraphs, and this semantic-level attention utilizes node features. Therefore, when aggregating heterogeneous subgraphs using attention mechanisms, it is possible to better distinguish the importance of different meta-paths.

In contrast, homogeneous subgraphs only contain nodes of the target type, so when aggregating with attention mechanisms, only the importance of individual nodes is considered, which does not adequately summarize the importance of different meta-paths. Therefore, compared to channel attention mechanisms, the strategy of guiding the fusion of homo-

geneous subgraphs with heterogeneous subgraphs under meta-paths is more effective.

5.10 Parameter analysis

In this section, we investigate the sensitivity of important parameters. We conduct a parameter analysis on the number of layers in the homogenous subgraphs and the number of layers in the heterogeneous subgraphs.

As shown in Fig. 9, the performance of node classification generally shows a trend of first increasing and then decreasing with the increase in the number of neural network layers. This is because when nodes aggregate information from their neighbors, the state updates of the nodes typically only consider information from one-hop neighbors. Therefore, the number of network layers reflects how many hops of neighbor information a node can integrate. During the training process, when the network layers are shallow, nodes may not be able to gather sufficient effective information, which can negatively impact classification performance. As the number of

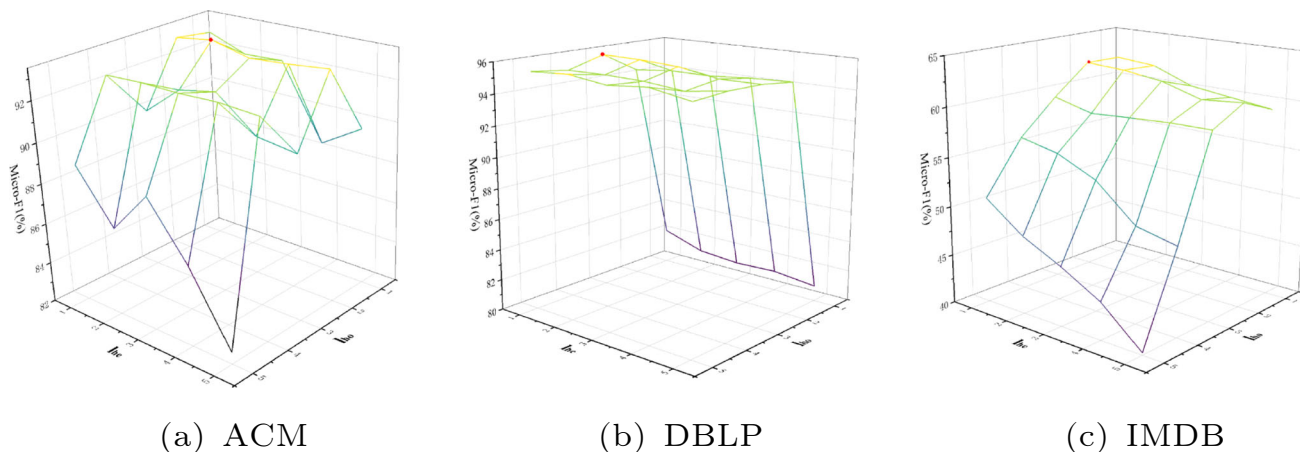


Fig. 9 Parameter sensitivity

network layers increases, nodes can integrate more effective information, thereby improving classification results. However, when the number of layers reaches a certain threshold, the nodes in the entire network may exhibit overly similar features, a phenomenon known as over-smoothing, which can lead to a decline in performance.

6 Conclusion

This paper proposes a semi-supervised heterogeneous graph contrastive learning model guided by label information, aiming to fully utilize label information and enrich the supervisory signal through contrastive learning. To address the first challenge, we construct a label graph, explicitly encoding label information into the learning process of the graph neural network, achieving joint representation learning of labels and nodes. To tackle the second challenge, when constructing positive and negative samples for graph contrastive learning, we introduce a method that jointly selects positive samples using both labels and meta-paths and utilizes contrastive loss to maximize the consistency between homogeneous and heterogeneous views. Extensive experiments conducted on various datasets fully demonstrate the superiority of the algorithm compared to others. Given the broad application prospects of heterogeneous graph neural network models, in the future, we will explore the construction of heterogeneous graph structures for legal judgment documents, as well as legal judgment prediction and legal text recommendation based on heterogeneous graph neural network models.

Acknowledgements This work is supported by National Key R&D Program of China (Grant No. 2022ZD0119501); the Natural Science Foundation of Shandong Province (Grant No. ZR2022MF268, ZR2021QG038); the Social Science Planning and Research Project of Shandong Province (Grant No.22CFXJ07), the ‘Qunxing Plan’ project of educational and teaching research of Shandong University of Science and Technology (Grant No. QX2020Z12), the Undergraduate Teaching Reform Research Project of Shandong Province (Grant No. M2023277).

Author Contributions Chao Li, Guoyi Sun, Juan Shan wrote the main manuscript text; Guoyi Sun and Xin Li prepared the result of our experiments; All authors reviewed the manuscript.

Availability of Supporting Data The datasets used in the experiments are publicly available in the online repository.

Declarations

Competing Interests The authors declare that there is no competing interests.

Ethical Approval Not applicable.

Consent to Participate There is the consent of all authors.

Human and Animal Ethics Not applicable.

Consent for publication There is the consent of all authors.

References

1. Wang Q, Zhu C, Zhang Y, Zhong H, Zhong J, Sheng VS (2022) Short text topic learning using heterogeneous information network. *IEEE Trans Knowl Data Eng* 35(5):5269–5281
2. Wang X, Bo D, Shi C, Fan S, Ye Y, Philip SY (2022) A survey on heterogeneous graph embedding: methods, techniques, applications and sources. *IEEE Transactions on Big Data*. 9(2):415–436
3. Han M, Zhang H, Li W, Yin Y (2023) Semantic-guided graph neural network for heterogeneous graph embedding. *Expert Syst Appl* 232:120810
4. Salamat A, Luo X, Jafari A (2021) Heterographrec: a heterogeneous graph-based neural networks for social recommendations. *Knowl-Based Syst* 217:106817
5. Huang M (2021) Research on graph network recommendation algorithm based on random walk and convolutional neural network. In: 2021 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS), pp 57–64. IEEE
6. Louis A, Van Dijk G, Spanakis G (2023) Finding the law: enhancing statutory article retrieval via graph neural networks. [arXiv:2301.12847](https://arxiv.org/abs/2301.12847)
7. Qi R, Zhang Z, Wu J, Dou L, Xu L, Cheng Y (2024) A new method for handling heterogeneous data in bioinformatics. *Comput Biol Med* 170:107937
8. Zhao J, Wang X, Shi C, Liu Z, Ye Y (2020) Network schema preserving heterogeneous information network embedding. In: International Joint Conference on Artificial Intelligence (IJCAI)
9. Yao K, Wang X, Li W, Zhu H, Jiang Y, Li Y, Tian T, Yang Z, Liu Q, Liu Q (2023) Semi-supervised heterogeneous graph contrastive learning for drug-target interaction prediction. *Comput Biol Med* 163:107199
10. Liu X, Zhang F, Hou Z, Mian L, Wang Z, Zhang J, Tang J (2021) Self-supervised learning: Generative or contrastive. *IEEE Trans Knowl Data Eng* 35(1):857–876
11. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. *Int conf learn represent*
12. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. *Adv neural inf process syst* 30
13. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y (2017) Graph attention networks. [arXiv:1710.10903](https://arxiv.org/abs/1710.10903)
14. Liao Z, Zhang X, Su W, Zhan K (2022) View-consistent heterogeneous network on graphs with few labeled nodes. *IEEE Trans Cyber*
15. You Y, Chen T, Sui Y, Chen T, Wang Z, Shen Y (2020) Graph contrastive learning with augmentations. *Adv Neural Inf Process Syst* 33:5812–5823
16. Zhao X, Wu J, Zhao X, Yin M (2023) Multi-view contrastive heterogeneous graph attention network for Incrna-disease association prediction. *Brief Bioinform* 24(1):548
17. Zhang Q, Zhao Z, Zhou H, Li X, Li C (2023) Self-supervised contrastive learning on heterogeneous graphs with mutual constraints of structure and feature. *Inf Sci* 640:119026
18. Xue W, He Z, Cui W, Li L, Yang Z, Lu S (2023) Unidirectional reflectionless propagation of near-infrared light in heterogeneous metamaterials. *Physica E* 147:115593
19. Wang X, Zhu M, Bo D, Cui P, Shi C, Pei J (2020) Am-gcn: adaptive multi-channel graph convolutional networks. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 1243–1253

20. Wang X, Ji H, Shi C, Wang B, Ye Y, Cui P, Yu PS (2019) Heterogeneous graph attention network. In: The world wide web conference, pp 2022–2032
21. Fu X, Zhang J, Meng Z, King I (2020) Magnn: metapath aggregated graph neural network for heterogeneous graph embedding. In: Proceedings of the web conference 2020, pp 2331–2341
22. Zhang C, Song D, Huang C, Swami A, Chawla NV (2019) Heterogeneous graph neural network. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 793–803
23. Hu Z, Dong Y, Wang K, Sun Y (2020) Heterogeneous graph transformer. In: Proceedings of the Web conference 2020, pp 2704–2710
24. Zhao J, Wang X, Shi C, Hu B, Song G, Ye Y (2021) Heterogeneous graph structure learning for graph neural networks. In: Proceedings of the AAAI conference on artificial intelligence, vol 35, pp 4697–4705
25. Yang Y, Guan Z, Li J, Zhao W, Cui J, Wang Q (2021) Interpretable and efficient heterogeneous graph convolutional network. *IEEE Trans Knowl Data Eng* 35(2):1637–1650
26. Zhang M, Wang X, Zhu M, Shi C, Zhang Z, Zhou J (2022) Robust heterogeneous graph neural networks against adversarial attacks. In: Proceedings of the AAAI conference on artificial intelligence, vol 36, pp 4363–4370
27. Ji H, Wang X, Shi C, Wang B, Philip SY (2021) Heterogeneous graph propagation network. *IEEE Trans Knowl Data Eng* 35(1):521–532
28. Liu Z, Wang C, Han C, Guo T (2023) Learning graph representation by aggregating subgraphs via mutual information maximization. *Neurocomputing* 548:126392
29. Fang U, Li J, Akhtar N, Li M, Jia Y (2023) Gomic: multi-view image clustering via self-supervised contrastive heterogeneous graph co-learning. *World Wide Web*. 26(4):1667–1683
30. Ren Y, Liu B, Huang C, Dai P, Bo L, Zhang J (2019) Heterogeneous deep graph infomax. *Workshop of deep learning on graphs: methodologies and applications co-located with the thirty-fourth AAAI conference on artificial intelligence*
31. Park C, Kim D, Han J, Yu H (2020) Unsupervised attributed multiplex network embedding. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 5371–5378
32. Wang X, Liu N, Han H, Shi C (2021) Self-supervised heterogeneous graph neural network with co-contrastive learning. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, pp 1726–1736

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



media, Heterogeneous graph neural networks, Natural language processing, Data mining and Network embedding learning etc.



Guoyi Sun received the bachelor degree in engineering from Linyi University (LYU), Linyi, China, in 2021. He is currently working towards the Master degree in the College of Electronic and Information Engineering, Shandong University of Science and Technology (SDUST). His research interest focuses on Heterogeneous graph neural networks.



Xin Li received the bachelor degree in engineering from Liaocheng University (LCU), Liaocheng, China, in 2022. She is currently working towards the Master degree in the College of Electronic and Information Engineering, Shandong University of Science and Technology (SDUST). Her research interest focuses on Heterogeneous graph neural networks and Graph structure learning.



Juan Shan associate professor, Master Supervisor, she received the B.S. from Yantai University (YTU), M.S from Shandong University of Science and Technology (SDUST), and received her Ph.D. degrees from Jilin University (JLU), she was a visiting scholar in China University of Political Science and Law From September 2012 to July 2013. Currently she works as an associate professor at Shandong University of Science and Technology. Dr. Shan is a director of the China Society of Private International Law. Her research interests include digital law, artificial intelligence law, international law etc.