IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 22 May 2022, accepted 16 June 2022, date of publication 29 June 2022, date of current version 22 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3187088

RESEARCH ARTICLE

Siamese Network Based Multiscale Self-Supervised Heterogeneous Graph Representation Learning

ZIJUN CHEN[1], LIHUI LUO[1], XUNKAI LI[1], BIN JIANG[1], QIANG GUO[2], (Member, IEEE), AND CHUNPENG WANG[1]

[1]School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai 264209, China
[2]School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan 250014, China

Corresponding author: Chunpeng Wang (wcp@sdu.edu.cn)

This work was supported in part by the Shandong Provincial Natural Science Foundation under Grant ZR2020MA064, in part by the National Natural Science Foundation of China under Grant 61873145, in part by the Natural Science Foundation of Shandong Province for Excellent Young Scholars under Grant ZR2017JL029, and in part by the Science and Technology Innovation Program for Distinguished Young Scholars of Shandong Province Higher Education Institutions under Grant 2019KJN045.

ABSTRACT Owing to label-free modeling of complex heterogeneity, self-supervised heterogeneous graph representation learning (SS-HGRL) has been widely studied in recent years. The goal of SS-HGRL is to design an unsupervised learning framework to represent complicated heterogeneous graph structures. However, based on contrastive learning, most existing methods of SS-HGRL require a large number of negative samples, which significantly increases the computation and memory costs. Furthermore, many methods cannot fully extract knowledge from a heterogeneous graph. To learn global and local information simultaneously at low time and space costs, we propose a novel **S**iamese **N**etwork based **M**ulti-scale bootstrapping contrastive learning approach for **H**eterogeneous graphs (SNMH). Specifically, we first obtain views under the meta-path schema and the 1-hop relation type schema through dual-schema view generation. Then, we propose cross-schema and cross-view bootstrapping contrastive objectives to maximize the similarity of node representations between different schemas and views. By integrating and optimizing the above objectives, we can extract local and global information and eventually obtain the node representations for downstream tasks. To demonstrate the effectiveness of our model, we conduct experiments on several public datasets. Experimental results show that our model is superior to the state-of-the-art methods on the premise of lower time and space complexity. The source code and datasets are publicly available at https://github.com/lorisky1214/SNMH.

INDEX TERMS Heterogeneous graph, graph representation learning, self-supervised learning, Siamese network, multiscale.

I. INTRODUCTION

As a method of analyzing and mining rich information in graph structural data, graph representation learning (GRL) has attracted wide attention in recent years. GRL aims to learn the high-order embeddings of nodes or graphs that preserve the information of node attributes and graph topological structure, which can be used for a wide variety of downstream tasks. Benefiting from the development of deep learning, most successful GRL methods extend neural networks to graph data, classified as graph neural networks (GNNs). They have obtained significant results on many tasks, such as node classification [3]–[5], recommendation system [6]–[8], and link prediction [9]–[11].

Despite the fruitful progress, GNNs are mostly applied in a supervised manner [3], [4], [12], [13], which requires a large number of labeled nodes for training. Moreover, the acquisition of label information in the real world is very costly and

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Tucci.

often requires domain-specific expertise, which is difficult to obtain. To address these problems, self-supervised GRL is presented, aiming to spontaneously extract supervised signals from the data itself without any label information. Following the concept of mutual information (MI) [18], a series of contrastive learning methods have recently achieved promising results in learning representations by contrasting positive and negative samples. For example, DGI [19] maximizes MI between the global summary representation and the local patches to generate a low-dimensional representation of each node using the embedding framework in the homogeneous graph, such as GCN [13]. On top of DGI, MVGRL [20] introduces multiple views to learn the representations of nodes and graphs by maximizing MI between the node representation of a view and the graph representation of the other view. After graph augmentation, GRACE [21] maximizes the similarity between the representations of the nodes on the two random perturbed views of the same original graph. Adopting different graph augmentation methods, GROC [22] improves robustness to adversarial attacks, and GCA [23] improves adaptive capacity. It is worth noting that all of the above methods are applied in homogeneous graphs that contain only one type of nodes and relations.

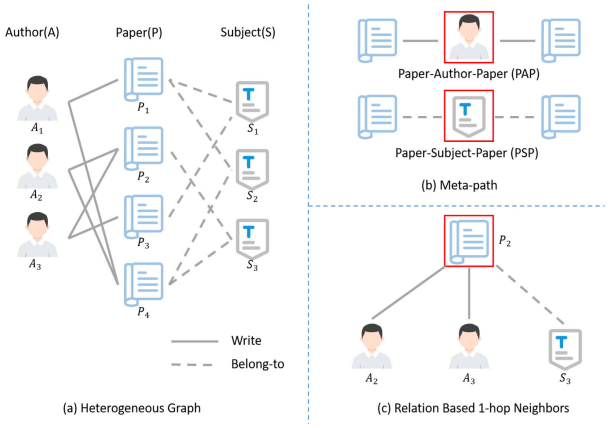

FIGURE 1. An example of a heterogeneous graph from the ACM dataset and relative illustrations of meta-path and relation based 1-hop neighbors. Nodes with red frames indicate that information is discarded during the encoding process.

However, real-world graphs often contain multiple node types and relation types represented by edges, which are called heterogeneous graphs with more comprehensive information and richer semantics. As a typical characteristic of heterogeneous graphs, meta-path [24] can capture semantic information in a graph by representing the composite relation between two nodes. Fig. 1 shows an example of a heterogeneous graph from the ACM dataset, which contains three types of nodes (Author, Paper and Subject). There are Write and Belong-to relation types between them. Meanwhile, the meta-paths between two papers can be divided into two types, i.e., Paper-Author-Paper (PAP) and Paper-Subject-Paper (PSP). PAP means that two papers belong to the same author, and PSP means the same subject. DMGI [25] and HDGI [26], two current self-supervised heterogeneous GRL methods, generate node embeddings for each meta-path type first and then integrate the embeddings with different semantic information using a consistent regularization framework. HeCo [27] conducts a further step by proposing a collaborative contrastive learning mechanism that encodes nodes to handle heterogeneity from both network schema and meta-path views. Although the aforementioned methods have achieved significant success, they are all subject to one of the following problems: 1) Ignoring local neighborhood information. If we only focus on semantic information, the representations will fail to extract useful information from direct neighbors. With this end, it is necessary to design a mechanism to simultaneously learn the rich local and global information in a graph. 2) Dependance on a large number of negative samples. In this case, it leads to high time and space complexity. At the same time, it is difficult for graphs to define negative samples in a principled way.

To solve the aforementioned problems, inspired by bootstrapping in the Siamese network [1], we propose SNMH, a novel multi-scale self-supervised heterogeneous GRL method to comprehensively extract rich information from heterogeneous graphs at low time and space costs. Specifically, distinct from current methods, we propose dual-schema view generation to obtain the meta-path based views and relation type based 1-hop views, which represent global and local information, respectively. Furthermore, we construct cross-view and cross-schema variant Siamese architectures. By maximizing the similarity of node representations between different views under the same schema and between two schemas, we can obtain node representations containing abundant node attributes and topological information without negative samples. Experimental results on various datasets demonstrate the excellent performance of our model.

Our contributions can be summarized as follows:

- SNMH is the first trial to apply bootstrapping in the Siamese network to self-supervised heterogeneous graph representation learning, which can reduce time and space costs by avoiding negative samples.
- The multi-scale optimization objective of SNMH, namely, cross-schema and cross-view contrastiveness, facilitates simultaneous learning of local and global information to ultimately obtain high-quality representations of nodes.
- We perform extensive experiments on different public datasets. Comparative results with state-of-the-art models demonstrate the superiority of our model with the premise of lower time and space complexity.

II. RELATED WORK

A. SIAMESE NETWORK

Siamese network is a network architecture that contains two identical structures. Initially, as a supervised learning method, it is often used on tasks such as forged signature detection [1] and face validation [2]. Recently, BYOL [30] introduces this structure into self-supervised visual representation learning,

which can provide results competing with state-of-the-art contrastive learning methods while avoiding using negative samples. BGRL [28] applies Siamese network to the GRL domain, which can be seen in Fig. 2. Through two different graph augmentations $\mathcal{A}_1$ and $\mathcal{A}_2$, two views can be obtained, based on which an online encoder ε_θ and a target encoder ε_ϕ are employed to generate node representations for each view. The predictor p_θ makes a prediction $\widetilde{\mathbf{Z}}_1$ of the target embedding $\widetilde{\mathbf{H}}_2$ utilizing the online embedding $\widetilde{\mathbf{H}}_1$. The ultimate objective is calculated as the similarity of $\widetilde{\mathbf{Z}}_1$ and $\widetilde{\mathbf{H}}_2$ with gradients flowing only via $\widetilde{\mathbf{Z}}_1$. The target parameters ϕ are updated to an exponential moving average (abbreviated as EMA) of θ. Among them, the additional predictor of the online encoder and the non-gradient descent update mechanism of the target encoder are the keys to achieving superior results for the Siamese network without using negative samples.

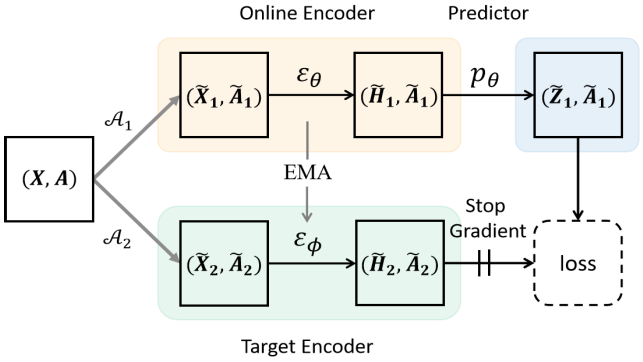

FIGURE 2. The architecture of the Siamese network applied in GRL. The orange, green and blue backgrounds indicate the online encoder, target encoder and prediction generated by the predictor, respectively (similar to Fig. 3 and Fig. 4).

B. SELF-SUPERVISED HETEROGENEOUS GRAPH REPRESENTATION LEARNING

For the self-supervised representation learning of heterogeneous graphs, most state-of-the-art methods follow a contrastive learning method, which learns low-dimensional features of nodes without labels by allowing the model to compare similar or different data points. For example, DMGI [25] and HDGI [26] utilize GNN encoders to learn node representations for each meta-path based view and then aggregate them through consensus regularization. Both methods conduct contrastive learning to learn node representations in heterogeneous graphs by maximizing mutual information between local patches and corresponding graph-level summaries of graphs. However, the limitation of these two methods is that they only consider the node attributes and global properties while ignoring the impact of local information on the quality of node embeddings. Recently, HeCo [27] proposes a collaborative contrastive mechanism to address this problem. HeCo learns node embeddings from network schema and meta-path views to capture both local and global structural information simultaneously. Nevertheless, despite the superior results of all the above methods, they all rely heavily on the number of negative samples, which greatly affects the computation and memory costs. Moreover, generating negative samples in a proper way is also a challenging task. To address the above problems, we propose a multi-scale self-supervised heterogeneous graph representation learning model, which aims to consider the rich information of node attributes as well as the local and global structures in heterogeneous graphs at lower computation and memory costs.

III. PRELIMINARY

A. HETEROGENEOUS GRAPH

A heterogeneous graph is defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is a set of nodes, and $\mathcal{E}$ is a set of edges. It has a node-type mapping function $\phi : \mathcal{V} \to \mathcal{T}$ and an edge-type mapping function $\psi : \mathcal{E} \to \mathcal{R}$, where $\mathcal{T}$ and $\mathcal{R}$ represent the node-type set and the edge-type set, respectively, and $|\mathcal{T}| + |\mathcal{R}| > 2$. Fig. 1 (a) shows an example of a heterogeneous graph with Author(A), Paper (P) and Subject (S) nodes. There are two types of relations, i.e., Write and Belong-to, which mean that the author writes the paper and the paper belongs to the subject, respectively.

In this paper, we represent the attributes of nodes with type ϕ_i as the initial feature matrix $\mathbf{X}_{\phi_i} \in \mathbb{R}^{|\mathcal{V}_{\phi_i}| \times F_{\phi_i}}$, where $|\mathcal{V}_{\phi_i}|$ is the number of nodes with type ϕ_i, and F_{ϕ_i} is the initial dimension. In this paper, we specify the set of target nodes as $\mathcal{V}_{\phi_t}$ for representation learning.

B. META-PATH

A meta-path Φ is defined as $v_1 \xrightarrow{R_1} v_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} v_{l+1}$ (abbreviated as $v_1 v_2 \cdots v_{l+1}$). It describes the composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between nodes v_1 and v_{l+1}, where $\circ$ represents a combination operator on the relation. Meta-paths can model rich semantic information in heterogeneous graphs. As shown in Fig. 1 (b), two papers can be connected by PAP and PSP meta-paths. PAP means that two papers belong to the same author, and PSP means the same subject.

In this paper, we represent the set of meta-paths as $\{\Phi_1, \Phi_2, \cdots, \Phi_P\}$, where P is the number of meta-path types. The topology of the view based on the meta-path type Φ_k can be expressed as $\mathbf{A}^{\Phi_k} \in \mathbb{R}^{|\mathcal{V}_{\phi_t}| \times |\mathcal{V}_{\phi_t}|}$. If there is a meta-path Φ_k between nodes v_i and v_j, then $\mathbf{A}_{ij}^{\Phi_k} = \mathbf{A}_{ji}^{\Phi_k} = 1$; otherwise $\mathbf{A}_{ij}^{\Phi_k} = \mathbf{A}_{ji}^{\Phi_k} = 0$.

C. RELATION BASED 1-HOP NEIGHBORS

For a node v, its relation based 1-hop neighbors can be represented as $\mathcal{S}_v = \{u : d(v, u) = 1\}$, where $d(v, u)$ is the shortest distance between nodes u and v. As shown in Fig. 1 (c), the relation based 1-hop neighbors of the central node P_2 are A_2, A_3 and S_3. They are directly connected, which reflects the local structural information in a heterogeneous graph.

Based on different relation types in a heterogeneous graph, we can construct $|\mathcal{R}|$ relation types based 1-hop views.

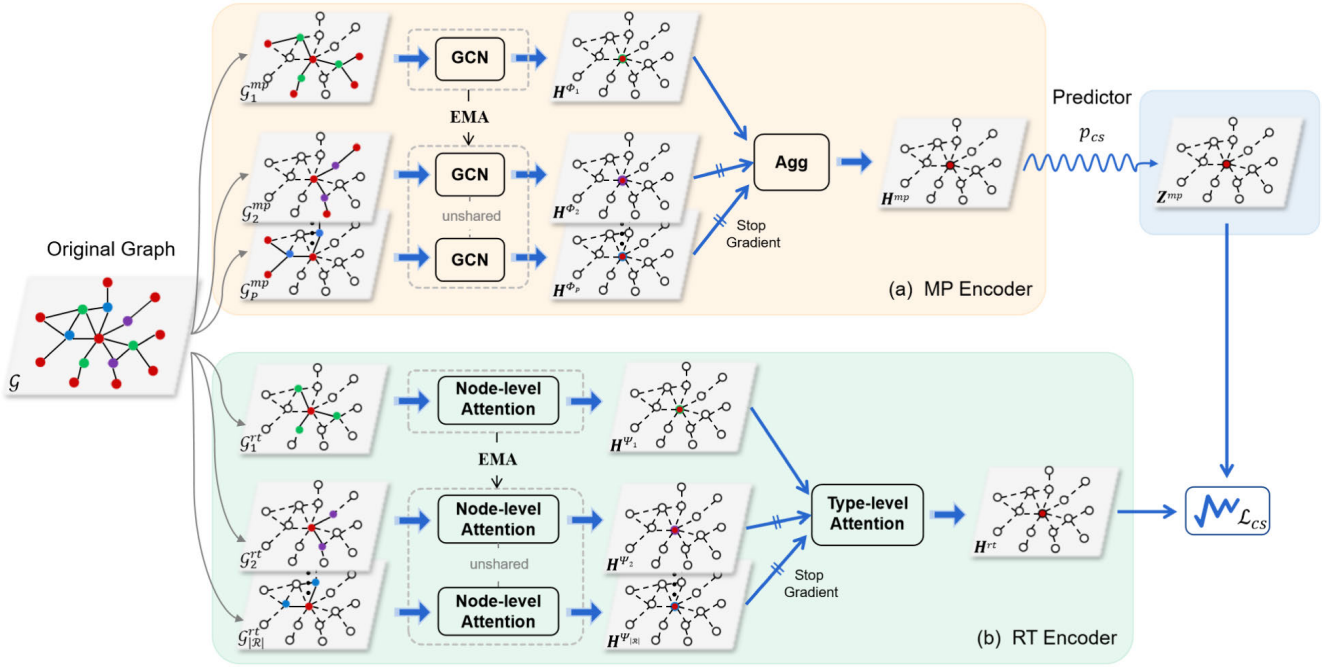

FIGURE 3. The architecture of SNMH's cross-schema mechanism.

We represent the neighborhood information pertaining to each view as a relation type based adjacency matrix $\mathbf{A}^{\Psi_k} \in \mathbb{R}^{|\mathcal{V}_{\phi_t}|\times|\mathcal{V}_{\phi_k}|}$. If there is an edge of relation type Ψ_k between target nodes $v_i \in \mathcal{V}_{\phi_t}$ and $v_j \in \mathcal{V}_{\phi_k}$, then $\mathbf{A}_{ij}^{\Psi_k} = 1$; otherwise, $\mathbf{A}_{ij}^{\Psi_k} = 0$.

IV. PROPOSED METHOD

We propose a Siamese network based self-supervised heterogeneous graph representation learning model with multi-scale optimization objectives. The cross-schema and cross-view architectures are shown in Fig. 3 and Fig. 4. First, we perform dual-schema view generation and obtain meta-path based views and relation type based 1-hop views. Afterward, we deploy variant Siamese networks between different views under the same schema and two schemas to construct a multi-scale bootstrapping contrastive learning mechanism. Finally, we systematically combine and optimize the cross-schema and cross-view optimization objectives to obtain the final node representations. In the following sections, we will introduce our model in detail.

A. DUAL-SCHEMA VIEW GENERATION

The factors affecting the representations of nodes in heterogeneous graphs are complex and contain the attributes of nodes, global information represented by meta-paths and local information represented by relation type based 1-hop neighbors. To simultaneously learn the above information, we innovatively propose a dual-schema view generation mechanism, as shown in Fig. 3.

Assume that the original graph is $\mathcal{G}$. To start with, according to the first schema, meta-path, we can generate a series of views:

$$\mathcal{G} = \{\mathcal{G}_1^{mp}, \mathcal{G}_2^{mp}, \cdots, \mathcal{G}_P^{mp}\}, \tag{1}$$

where $\mathcal{G}_i^{mp} = (\mathbf{X}_{\phi_t}, \mathbf{A}^{\Phi_i})$, $\mathbf{X}_{\phi_t} \in \mathbb{R}^{|\mathcal{V}_{\phi_t}|\times F_{\phi_t}}$ is the feature matrix of target nodes, and $\mathbf{A}^{\Phi_i} \in \{0,1\}^{|\mathcal{V}_{\phi_t}|\times|\mathcal{V}_{\phi_t}|}$ is the adjacency matrix of view $\mathcal{G}_i^{mp}$.

Next, for the second schema, i.e., the 1-hop relation type, we can still obtain a series of views:

$$\mathcal{G} = \{\mathcal{G}_1^{rt}, \mathcal{G}_2^{rt}, \cdots, \mathcal{G}_{|\mathcal{R}|}^{rt}\}, \tag{2}$$

where $\mathcal{G}_i^{rt} = (\mathbf{X}_{\phi_t}, \mathbf{X}_{\phi_i}, \mathbf{A}^{\Psi_i})$, $\mathbf{X}_{\phi_t}$ is the same as formula (1), and $\mathbf{X}_{\phi_i} \in \mathbb{R}^{|\mathcal{V}_{\phi_i}|\times F_{\phi_i}}$ is the 1-hop neighbors feature matrix based on relation type Ψ_i. $\mathbf{A}^{\Psi_i} \in \{0,1\}^{|\mathcal{V}_{\phi_t}|\times|\mathcal{V}_{\phi_i}|}$ is the adjacency matrix of this view, which generally is not a square matrix.

B. NODE FEATURE TRANSFORMATION

Since node types contained in heterogeneous graphs are not singular with different feature spaces or feature dimensions, we need to first project them into the same dimension space to facilitate the processing of subsequent models. Specifically, for the nodes of a certain type ϕ_i, we construct a specific type of linear transformation that transforms the initial features $\mathbf{X}_{\phi_i}$ of the nodes into a unified dimension. The specific formula is shown as follows:

$$\mathbf{X}'_{\phi_i} = \sigma(\mathbf{W}_{\phi_i} \cdot \mathbf{X}_{\phi_i} + \mathbf{b}_{\phi_i}), \tag{3}$$

where $\mathbf{X}'_{\phi_i} \in \mathbb{R}^{|\mathcal{V}_{\phi_i}|\times F}$, F is the node dimension after the feature transformation, $\sigma(\cdot)$ is an activation function, $\mathbf{W}_{\phi_i}$ is a linear transformation parameter matrix, and $\mathbf{b}_{\phi_i}$ is the bias.

C. CROSS-SCHEMA BOOTSTRAPPING CONTRASTIVENESS

In the learning mechanism of cross-schema bootstrapping contrastiveness, we design a variant Siamese structure. Unlike conventional Siamese networks, our proposed variation in the cross-schema learning mechanism has an online and target encoder with distinct structures. Our online encoder and target encoder are shown in Fig. 3 (a) and (b), respectively. To distinguish them from the conventional Siamese network and intuitively express their characteristics, we define them as meta-path guided encoder and 1-hop relation type guided encoder, respectively (abbreviated as MP encoder and RT encoder).

The cross-schema learning procedure is shown in Fig. 3, where $\mathbf{Z}^{mp} = p_{cs}(\mathbf{H}^{mp})$ is the matrix of predicted node representations obtained after the input of the output node representations of the MP encoder to its additional predictor p_{cs}, and $\mathbf{H}_{rt}$ is the output node representation matrix of the RT encoder. We form the cross-schema bootstrapping contrastiveness by maximizing the cosine similarity between the above two. We express the optimization objective as the following formula:

$$\mathcal{L}_{cs} = -\frac{2}{|\mathcal{V}_{\phi_t}|}\sum_{i=0}^{|\mathcal{V}_{\phi_t}|-1}\frac{\mathbf{Z}_i^{mp}\mathbf{H}_i^{rt\top}}{||\mathbf{Z}_i^{mp}||||\mathbf{H}_i^{rt}||}, \tag{4}$$

where $\mathbf{Z}_i^{mp} \in \mathbb{R}^{1\times d}$ and $\mathbf{H}_i^{rt} \in \mathbb{R}^{1\times d}$ are the representations of node i output by the predictor p_{cs} and RT encoder, respectively.

For the acquisition of $\mathbf{H}^{mp}$ and $\mathbf{H}^{rt}$, as well as the update of the MP encoder and RT encoder, we will specify them in detail in the next two subsections.

1) META-PATH GUIDED ENCODER

Here, we aim to learn the characteristics of nodes in the high-order meta-path schema. After Sections IV.A and IV.B, we obtain multiple views from feature-transformed nodes generated based on meta-path types. We represent these views as $(\mathbf{X}'_{\phi_t}, \mathbf{A}^{\Phi_i})$. To extract the node attributes and semantic information contained in each view, we plunge each of the above views into a GNN:

$$\mathbf{H}^{\Phi_i} = f_{\Phi_i}(\mathbf{X}'_{\phi_t}, \mathbf{A}^{\Phi_i}). \tag{5}$$

For f_{Φ_i}: $\mathbb{R}^{|\mathcal{V}_{\phi_t}|\times F} \times \mathbb{R}^{|\mathcal{V}_{\phi_t}|\times|\mathcal{V}_{\phi_t}|} \to \mathbb{R}^{|\mathcal{V}_{\phi_t}|\times d}$, we use a meta-path specific single-layer GCN [13]. The node embedding matrix encoded by the GCN can be expressed as:

$$\mathbf{H}^{\Phi_i} = \sigma(\tilde{\mathbf{D}}^{\Phi_i\,-\frac{1}{2}}\tilde{\mathbf{A}}^{\Phi_i}\tilde{\mathbf{D}}^{\Phi_i\,-\frac{1}{2}}\mathbf{X}'_{\phi_t}\mathbf{W}^{\Phi_i}), \tag{6}$$

where $\tilde{\mathbf{A}}^{\Phi_i} = \mathbf{A}^{\Phi_i} + \mathbf{I}$, $\mathbf{I}$ is an identity matrix, $\tilde{\mathbf{D}}^{\Phi_i}$ is the degree matrix of $\tilde{\mathbf{A}}^{\Phi_i}$, and $\mathbf{W}^{\Phi_i} \in \mathbb{R}^{F\times d}$ is a weight matrix that is not shared between different views. σ is a nonlinear activation function, and here, we use PReLU.

Each GCN encoder encodes a node embedding under a meta-path. Apparently, the effects of different meta-paths on the quality of the resulting node embeddings are distinct. Intuitively, if target nodes are mostly connected through a certain type of meta-path, this meta-path type affects their representations most. Based on this, we treat the encoder of the view with the largest number of meta-paths contained as the anchor encoder. As shown in Fig. 3 (a), we assume that the top encoder is the anchor encoder, and the bottom encoders are non-anchor encoders. The ellipses in the figure indicate that the number of non-anchor encoders may be 1, 2, 3..., which depends on the number of meta-path types in a heterogeneous graph (we assume a minimum number of 2).

During the learning process, only the parameters of the anchor encoder are updated by gradient descent to reduce the target loss, while the parameters of other non-anchor encoders follow different targets. The intuition behind this is that the slow-moving non-anchor encoders act as a stabilizer to encode the meta-paths that are not the most influential. This guides the anchor encoder to learn to explore richer and better representations on the basis of the most influential meta-paths without relying on additional negative samples to avoid a collapse. The parameters of the non-anchor encoders are updated as an EMA of the parameters of the anchor encoder:

$$\delta = \tau\cdot\delta + (1-\tau)\cdot\eta, \tag{7}$$

where η and δ are the parameters of the anchor encoder and non-anchor encoders respectively. τ is a decay rate that controls the distance between η and δ, and its update can be seen in formula (19).

After the above meta-path specific node representation learning, we obtain a set of node embeddings $\{\mathbf{H}^{\Phi_i}\}_{i=1}^{P}$.

Now, we need to aggregate the node embeddings above. Considering that the appropriate aggregation methods may change for datasets with different distributions of the number of meta-paths, we implement distinct aggregation methods for different datasets, which are shown as follows:

a: AVERAGE POOLING

The first aggregation method is average pooling, which calculates the average of the set of embedding matrices:

$$\mathbf{H}^{mp} = \frac{1}{P}\sum_{i=1}^{P}\mathbf{H}^{\Phi_i}. \tag{8}$$

b: SEMANTIC-LEVEL ATTENTION

For the second method, we employ semantic-level attention [31] to fuse the node embeddings into the final embedding $\mathbf{H}^{mp}$ in the meta-path schema:

$$\mathbf{H}^{mp} = \sum_{i=1}^{P}\beta_{\Phi_i}\cdot\mathbf{H}^{\Phi_i}, \tag{9}$$

where β_{Φ_i} weighs the importance of the meta-path Φ_i, which is calculated as follows:

$$e^{\Phi_i} = \frac{1}{|\mathcal{V}_{\phi_t}|} \sum_{n=0}^{|\mathcal{V}_{\phi_t}|-1} \mathbf{q}_{\Phi_i}^{\top} \cdot tanh(\mathbf{W}_{mp} \cdot \mathbf{H}_n^{\Phi_i} + \mathbf{b}_{mp}),$$
$$\beta_{\Phi_i} = softmax(e^{\Phi_i}) = \frac{exp(e^{\Phi_i})}{\sum_{j=1}^{P} exp(e^{\Phi_j})}, \tag{10}$$

where $\mathbf{W}_{mp}$ and $\mathbf{b}_{mp}$ are the learnable parameters. $\mathbf{q}_{\Phi_i}$ denotes the semantic-level attention vector.

We will show details about the correspondence between datasets and aggregation methods in Section V.F. The final node embedding $\mathbf{H}^{mp}$ in the meta-path schema after aggregation will be fed into the predictor p_{cs}, and then $\mathbf{Z}^{mp}$ is generated.

2) 1-HOP RELATION TYPE GUIDED ENCODER

As mentioned above, we can also obtain multiple relation type based 1-hop views with feature-transformed nodes. In each view, the target node is connected to its 1-hop neighbors through the edges of a specific relation type. It is obvious that different types of neighbors contribute differently to the node embeddings, and different nodes of the same type are also different. Therefore, in the RT encoder, we adopt a hierarchical attention mechanism, i.e., node- and type-level attention.

After the node feature transformation, each of the relation type based 1-hop views can be represented as $(\mathbf{X}'_{\phi_t}, \mathbf{X}'_{\phi_i}, \mathbf{A}^{\Psi_i})$, where ϕ_i represents the type of 1-hop neighbors corresponding to the relation type Ψ_i. To extract the local structural information of the nodes in a graph, we feed each of the views into an identically structured GNN:

$$\mathbf{H}^{\Psi_i} = g_{\Psi_i}(\mathbf{X}'_{\phi_t}, \mathbf{X}'_{\phi_i}, \mathbf{A}^{\Psi_i}). \tag{11}$$

Unlike the MP encoder, we set g_{Ψ_i} as a node-level attention layer here. For node n in the Ψ_i relation type view, its representation in this layer can be calculated as follows:

$$\mathbf{H}_n^{\Psi_i} = \sigma\left(\sum_{m \in \mathcal{N}_n^{\Psi_i}} \alpha_{n,m}^{\Psi_i} \cdot \mathbf{X'}_m^{\Psi_i} \right), \tag{12}$$

where $\mathcal{N}_n^{\Psi_i}$ is the set of 1-hop neighbors of node n defined by Ψ_i, $\mathbf{X'}_m^{\Psi_i}$ denotes the feature vector of node m, and σ is a nonlinear activation function. Here, we use LeakyReLU. $\alpha_{n,m}^{\Psi_i}$ measures the importance of node m to node n, which can be calculated as follows:

$$\alpha_{n,m}^{\Psi_i} = \frac{exp\left(LeakyReLU\left(\mathbf{a}_{\Psi_i}^{\top} \cdot [\mathbf{X'}_n^{\Psi_i} || \mathbf{X'}_m^{\Psi_i}]\right)\right)}{\sum_{j \in \mathcal{N}_n^{\Psi_i}} exp\left(LeakyReLU\left(\mathbf{a}_{\Psi_i}^{\top} \cdot [\mathbf{X'}_n^{\Psi_i} || \mathbf{X'}_j^{\Psi_i}]\right)\right)}, \tag{13}$$

where $\mathbf{a}_{\Psi_i} \in \mathbb{R}^{2F \times 1}$ is the node-level attention vector and $||$ indicates the concatenating operation.

For the selection of nodes in $\mathcal{N}_n^{\Psi_i}$, we do not simply delimit all the nodes directly connected with node n through Ψ_i. Instead, we design a threshold Γ_{Ψ_i}. When the number of neighbors corresponding to Ψ_i is greater than the specified Γ_{Ψ_i}, we will non-repeatedly choose Γ_{Ψ_i} neighbors at random to join $\mathcal{N}_n^{\Psi_i}$. Otherwise, the Γ_{Ψ_i} neighbors can be selected repeatedly. In this way, the threshold ensures that each node under the same view aggregates the same amount of neighborhood information, while the random selection ensures the diversity of node embeddings in each epoch.

The specific learning process of each view is similar to the meta-path. The only difference is that we choose the view with the largest number of 1-hop neighbors as the anchor encoder here.

After learning the 1-hop relation type specific node representations described above, we obtain a set of node embeddings $\{\mathbf{H}^{\Psi_i}\}_{i=1}^{|\mathcal{R}|}$. Next, we use type-level attention to fuse them together to obtain the final embedding $\mathbf{H}^{rt}$ in the 1-hop relation type schema:

$$\mathbf{H}^{rt} = \sum_{i=1}^{|\mathcal{R}|} \mu_{\Psi_i} \cdot \mathbf{H}^{\Psi_i}, \tag{14}$$

μ_{Ψ_i} weighs the importance of the 1-hop relation type Ψ_i, which can be calculated as follows:

$$u^{\Psi_i} = \frac{1}{|\mathcal{V}_{\phi_t}|} \sum_{n=0}^{|\mathcal{V}_{\phi_t}|-1} \mathbf{q}_{\Psi_i}^{\top} \cdot tanh(\mathbf{W}_{rt} \cdot \mathbf{H}_n^{\Psi_i} + \mathbf{b}_{rt}),$$
$$\mu_{\Phi_i} = softmax(u^{\Phi_i}) = \frac{exp(u^{\Phi_i})}{\sum_{j=1}^{|\mathcal{R}|} exp(u^{\Phi_j})}, \tag{15}$$

where $\mathbf{W}_{rt}$ and $\mathbf{b}_{rt}$ are learnable parameters. $\mathbf{q}_{\Psi_i}$ is the type-level attention vector.

D. CROSS-VIEW BOOTSTRAPPING CONTRASTIVENESS

In addition to the bootstrapping contrastiveness between the two schemas, we additionally consider the relationship between the views within the meta-path schema, which acts as a strong regularization and is highly informative for improving the performance of our model. Details are shown in Fig. 4.

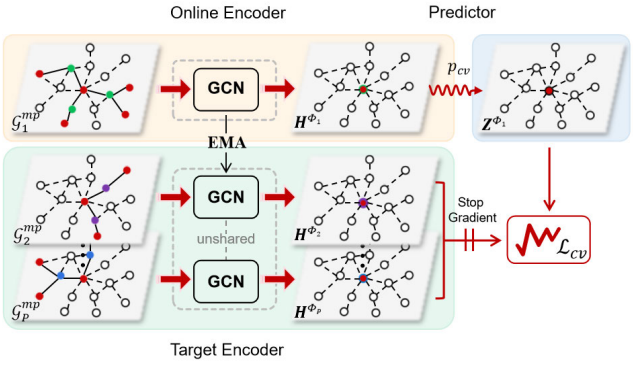

FIGURE 4. The architecture of SNMH's cross-view mechanism.

Since there are usually more than two meta-path types in a heterogeneous graph, more than two encoders with the

TABLE 1. The statistics of the datasets used in our experiments.

Dataset	Node type	# Nodes	Relation type	# Relations	Meta-path	# Classes
ACM	Paper (P) Author (A) Subject (S)	4019 7167 60	Paper-Author Paper-Subject	13407 4019	PAP PSP	3
DBLP	Author (A) Paper (P) Conference (C) Term (T)	4057 14328 20 7723	Author-Paper Paper-Conference Paper-Term	19645 14328 85810	APA APCPA APTPA	4
Freebase	Movie (M) Actor (A) Direct (D) Writer (W)	3492 33401 2502 4459	Movie-Actor Movie-Direct Movie-Writer	65341 3762 6414	MAM MDM MWM	3

same structure but different parameters are needed to encode different views inside the same schema. Therefore, unlike the traditional Siamese network that contains only one online encoder and one target encoder, we set the encoders according to different meta-paths. We consider the encoder of the view with the most meta-paths as an online encoder (connected to an additional predictor) and the rest as target encoders.

Assume that the meta-path corresponding to the online encoder is Φ_1. In each epoch, the online encoder generates an online representation $\mathbf{H}^{\Phi_1}$. Similarly, other target encoders also generate a series of representations $\{\mathbf{H}^{\Phi_i}\}_{i=2}^{P}$. Then, $\mathbf{H}^{\Phi_1}$ is fed into a predictor p_{cv} to generate $\mathbf{Z}^{\Phi_1} = p_{cv}(\mathbf{H}^{\Phi_1})$.

The parameters of the online encoder are updated by maximizing the similarity between the predictor's prediction and each target representation, following a gradient of cosine similarity:

$$\mathcal{L}_{cv} = -\frac{2}{|\mathcal{V}_{\phi_t}| \cdot (P-1)} \sum_{j=2}^{P} \sum_{i=0}^{|\mathcal{V}_{\phi_t}|-1} \frac{\mathbf{Z}_i^{\Phi_1} \mathbf{H}_i^{\Phi_j \mathsf{T}}}{||\mathbf{Z}_i^{\Phi_1}|| ||\mathbf{H}_i^{\Phi_j}||}. \quad (16)$$

During training, the parameters of the target encoders do not receive the gradient directly but are updated to the EMA of the online encoder by formula (7), where η and δ are the parameters of the online encoder and target encoders, respectively.

E. MULTI-SCALE MODEL TRAINING

To learn the attributes and local structural information, as well as the global topological information simultaneously, we systematically fuse the cross-schema and cross-view optimization objectives. We define the overall objective as follows:

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{cv} + (1-\lambda) \cdot \mathcal{L}_{cs}, \quad (17)$$

λ is a hyperparameter used to weigh these two objectives. According to the different effects obtained by setting different λ, we can determine the impact of meta-paths and relation based 1-hop neighbors (representing global and local information, respectively) on node representations for a specific dataset.

We train our proposed model end-to-end by optimizing the above objective. Finally, we obtain the node representations used for downstream tasks.

TABLE 2. The specific characteristics of baselines (X: Learning with initial features, A: Learning with the adjacency matrix, Y: Learning with node labels, HG: Suitable for heterogeneous graph).

Method	X	A	Y	HG
Raw	✓	×	×	×
HERec	×	✓	×	✓
HAN	✓	✓	✓	✓
DGI	✓	✓	×	×
DMGI	✓	✓	×	✓
SNMH	✓	✓	×	✓

V. EXPERIMENT

A. DATASETS

To evaluate the performance of SNMH, we conduct experiments with three public datasets, and the basic information statistics of the datasets are shown in Table 1.

- **ACM**: This is an academic paper dataset. The target nodes are papers, which are divided into three classes, namely, database, wireless communication and data mining. The initial features of papers are the bag-of-words representation of keywords. There are two meta-paths defined in ACM, including Paper-Author-Paper (PAP) and Paper-Subject-Paper (PSP) [32].
- **DBLP**: This is another paper dataset whose target nodes are authors. These nodes can be divided into four classes, including database, data mining, machine learning and information retrieval. We extract three meta-paths from the graph with Author-Paper-Author (APA), Author-Paper-Conference-Paper-Author (APCPA) and Author-Paper-Term-Paper-Author (APTPA) [33].
- **Freebase**: This is a dataset of movies. There are three classes of target movie nodes, namely action, comedy and drama. We encode the initial feature of the target node as a one-hot vector. In the experiments, the meta-paths we consider are Movie-Actor-Movie (MAM), Movie-Direct-Movie (MDM) and Movie-Writer-Movie (MWM) [34].

B. BASELINES

We implement five methods as baselines, and their specific characteristics are shown in Table 2, where Raw indicates that the initial features of the target nodes are treated as embeddings. HERec [36] is an unsupervised heterogeneous method that only uses topology structure to learn. HAN [31]

TABLE 3. Experimental results (% ± σ) on node classification.

Method	ACM			DBLP			Freebase		
	AUC	MaF1	MiF1	AUC	MaF1	MiF1	AUC	MaF1	MiF1
Raw	90.81 ± 0.1	76.48 ± 0.0	75.97 ± 0.1	84.98 ± 0.0	67.67 ± 0.1	68.03 ± 0.1	50.02 ± 0.2	32.57 ± 0.2	35.42 ± 0.2
HERec	81.64 ± 0.7	64.35 ± 0.8	65.15 ± 0.9	97.93 ± 0.1	89.73 ± 0.4	90.15 ± 0.4	73.89 ± 0.4	55.78 ± 0.5	57.92 ± 0.5
HAN	94.68 ± 1.4	88.41 ± 1.1	88.10 ± 1.2	97.48 ± 0.6	88.87 ± 1.0	89.47 ± 0.9	73.26 ± 2.1	53.16 ± 2.8	57.24 ± 3.2
DGI	91.41 ± 1.9	80.03 ± 3.3	80.15 ± 3.2	97.12 ± 0.4	88.62 ± 0.6	89.22 ± 0.5	72.80 ± 0.6	54.90 ± 0.7	58.16 ± 0.9
DMGI	96.79 ± 0.2	87.97 ± 0.4	87.82 ± 0.5	97.23 ± 0.2	89.25 ± 0.4	89.92 ± 0.4	73.19 ± 1.2	55.79 ± 0.9	58.26 ± 0.9
SNMH	**96.82 ± 0.1**	**88.48 ± 0.5**	**88.30 ± 0.5**	**98.00 ± 0.2**	**89.88 ± 0.9**	**90.37 ± 0.9**	**74.18 ± 1.7**	**56.30 ± 0.9**	**58.44 ± 0.6**

is a semi-supervised heterogeneous method that extracts node attributes and structural information in the graph through node- and semantic-level attention. DGI [19] is an unsupervised homogeneous method, while DMGI [25] is an unsupervised heterogeneous method. The inputs of these two models all contain the initial features and adjacency matrices of nodes.

C. EXPERIMENTAL SETUP

We implement SNMH using PyTorch. The model parameters are initialized by Glorot Initialization [37], and Adam [35] is used as the optimizer. We set the initial learning rate as $\gamma_0 = 0.5$ and the number of total epochs as $n_{total} = 10000$. To increase stability, we use batch normalization between layers and learning rate with a cosine schedule [28], which can be expressed by the following formula:

$$\gamma_i \triangleq \begin{cases} \frac{i \times \gamma_0}{n_{split}}, & i \leqslant n_{split} \\ \gamma_0 \times (1 + cos\frac{(i - n_{split}) \times \pi}{n_{total} - n_{split}}) \times 0.5, & n_{split} \leqslant i \leqslant n_{total}. \end{cases} \tag{18}$$

In all experiments, we fix $n_{split} = 1000$. For each dataset, patience is set with a range of { 10, 20, 30 }, i.e., training will be terminated when the loss does not decrease for consecutive patience epochs. To prevent overfitting, we set specific dropout values to each dataset and perform on both the feature vectors and the attention vectors. For simplicity, both predictors p_{cv} and p_{cs} are fixed as an MLP with a single hidden layer, whose dimension is set to 128. In the cross-schema and cross-view learning mechanisms, parameters updated by EMA are initialized following the same distribution as other parameters updated by gradient descent. The decay rate τ in formula (7) is initialized as $\tau_0 = 0.99$, which also follows a cosine schedule to update:

$$\tau_i \triangleq 1 - \frac{1 - \tau_0}{2} \times (cos(\frac{i \times \pi}{n_{total}}) + 1). \tag{19}$$

Unlike the target nodes of ACM and DBLP that have original features which can be used directly, the target nodes of Freebase do not have original features, and its initial feature matrix is defined as a sparse matrix with ones on the diagonal. We set the embedding dimensions of the target nodes of ACM, Freebase and DBLP to 64, 128 and 256, respectively. We run 10 times randomly and present the average results with the standard deviation values.

In the comparative experiments, we mainly refer to HeCo [27]. For HERec, we set the window size, the number of walks per node and the walk length to 5, 40 and 100, respectively. For both HERec and DGI, we test all meta-paths and present their best results. Other parameters follow the settings in the original paper.

D. NODE CLASSIFICATION

To evaluate the trained graph encoder, we use the learned node embeddings to fit a logistic regression classifier. During the fitting process, the embeddings are frozen to prevent any gradient flow back to the encoder. For the Freebase, DBLP and ACM datasets, we randomly select 20, 40 and 60 labeled nodes in each class as the training set, respectively, and 1000 nodes as the validation set and 1000 as the test set. We present the test performance when the validation set presents the optimal result. We compare SNMH with other baselines by AUC, Micro-F1 and Macro-F1. The results are shown in Table 3, where we mark the best performance in bold. As shown in the table, SNMH outperforms all datasets than other baselines. We attribute the results to the following two points: 1) We design a novel network based on the Siamese network, which can extract the information in historical representations without relying on negative samples. 2) We construct a multi-scale mechanism to learn node features as well as global and local structural information in heterogeneous graphs from cross-view and cross-schema perspectives. Even if the label information of nodes is used in the training process of HAN, SNMH is still superior to HAN, which also confirms the effectiveness of the self-supervised learning of SNMH.

E. VISUALIZATION

To intuitively evaluate our model, we visualize the node embeddings of ACM obtained by HERec, HAN, DMGI and SNMH using the t-SNE [38] algorithm, and the results are shown in Fig. 5. We also compute the Silhouette scores for different methods, which are 0.209, 0.323, 0.327 and 0.335. We can see that HERec cannot effectively identify different classes because of the lack of initial features. Even if HAN takes node labels as input, SNMH still works better than HAN. SNMH has a clearer boundary and denser clusters

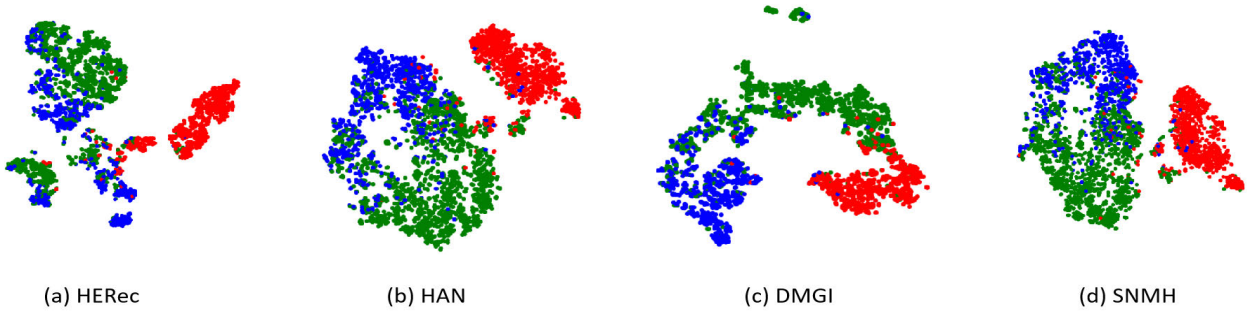

FIGURE 5. Visualization of the node embeddings of ACM, where different colors indicate different classes.

TABLE 4. Time and space cost comparison between SNMH and HeCo.

Dataset	Time (s)		Graphics Memory (GiB)	
	SNMH	HeCo	SNMH	HeCo
ACM	111.6	163.0	1.92	2.21
DBLP	119.6	215.2	2.90	3.23
Freebase	214.3	887.1	5.98	6.16

than the others, as well as a higher Silhouette score, which demonstrate the effectiveness of our method.

F. ANALYSIS

1) ANALYSIS OF TIME AND SPACE COSTS

We conduct comparative experiments in the server with an NVIDIA GeForce RTX 2080 GPU to analyze the time and space superiority of SNMH. To ensure consistency of experimental conditions, we choose HeCo as the comparison method. Similar to SNMH, HeCo also uses the initial feature and adjacency matrix as input and encodes both meta-paths and 1-hop neighbors, ensuring that the time and space costs are only affected by the model itself. We employ SNMH and HeCo to perform the same number of experiments on each dataset and then average the running time and graphics memory after removing a maximum and minimum value. The results of the comparative experiments are shown in Table 4.

As shown in Table 4, SNMH uses less running time and graphics memory than HeCo on all three datasets under the same experimental conditions. Therefore, we can conclude that SNMH can reduce time and space costs, which facilitates the implementation of our experiments.

To further illustrate the time and space superiority of our model in a variety of situations, we conduct experiments using SNMH and HeCo at different values of the threshold Γ_{Ψ_i}, i.e., the number of sampled 1-hop neighbors of target nodes. The results are shown in Fig. 6, where the solid lines represent graphics memory and the dashed lines represent time. Because each paper belongs to only one subject (S) in ACM, we only adjust the number of sampled A-type nodes. We can see that the changing trends of time and space are similar for both datasets. SNMH requires a stable amount of time and is always substantially less than HeCo. As the threshold is raised, both SNMH and HeCo's graphics memory rises slowly. These findings indicate SNMH's advantage in terms of time and space when sampling more 1-hop neighbors of target nodes.

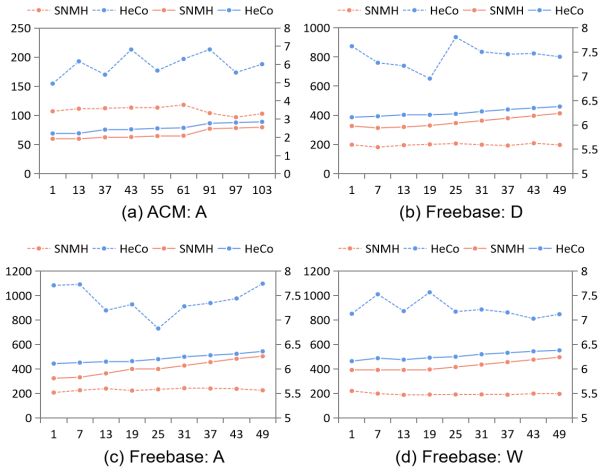

FIGURE 6. Time and space costs when changing the number of sampled 1-hop neighbors. The x-axis represents the number of sampled 1-hop neighbors. The y-axis on the left represents time (s), and the y-axis on the right represents graphics memory (GiB).

In addition, we perform experiments to see how the number of meta-paths affects time and space. The results are shown in Fig. 7, where the solid lines represent graphics memory and the dashed lines represent time. It can be seen from the figure that the curves representing different meta-paths show almost the same changing trend for time, and they are all much smaller than HeCo, indicating that the number of meta-paths has minimal influence on the time required for SNMH. When the number of MAM increases, the graphical memory required for SNMH increases slowly and is always less than HeCo, but it remains constant when the number of MDM and MWM increases. This finding suggests that MAM, as the meta-path encoded by the online encoder (anchor encoder), has a greater impact on the space required by the model. These results also prove the superiority of SNMH in time and space when more meta-paths are encoded.

2) ANALYSIS OF AGGREGATING META-PATH SPECIFIC NODE REPRESENTATIONS

From Section IV.C, there are two methods to aggregate meta-path specific node representations, including average

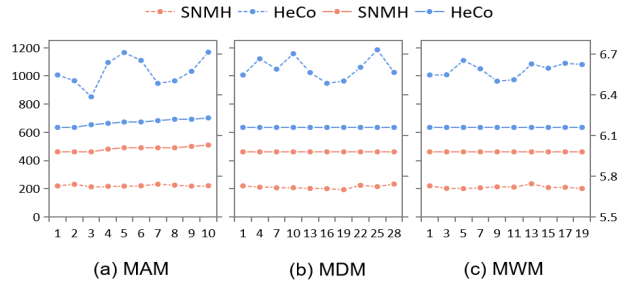

FIGURE 7. Time and space costs when changing the number of meta-paths. The x-axis represents the multiples of the original number of meta-paths. The y-axis on the left represents time (s), and the y-axis on the right represents graphics memory (GiB).

pooling and semantic-level attention. To explore the influence of different aggregation methods on the quality of the final node embeddings, we conduct experiments on ACM, DBLP and Freebase, and the results are shown in Fig. 8.

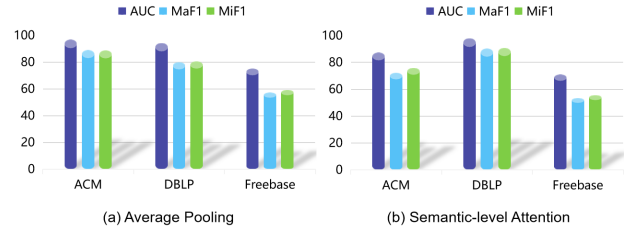

FIGURE 8. Analysis of methods for aggregating meta-path specific node representations.

From the figure, we conclude that different datasets are suitable for different aggregation methods. For ACM and Freebase, it is better to use average pooling, while DBLP is more suitable for semantic-level attention. In other words, identifying differences in meta-paths is more crucial for datasets such as DBLP, where the number of different meta-paths varies substantially. In addition, different datasets have distinct sensitivities to aggregation methods. ACM is obviously more affected than DBLP and Freebase when adopting different aggregation methods.

3) ANALYSIS OF HYPERPARAMETERS

In this section, we also investigate a key hyperparameter, i.e., the equilibrium parameter λ in the final objective (17). We perform node classification on ACM, DBLP and Freebase, and then show AUC, Macro-F1 and Micro-F1 values under different λ, as shown in Fig. 9.

It can be seen from the figure that all three datasets practically approach a minimum at $\lambda = 1$, demonstrating the necessity for us to consider both meta-paths and relation based 1-hop neighbors using the cross-schema mechanism. Furthermore, at $\lambda = 0$ (i.e., only the cross-schema part of the model remains), DBLP obtains the minimum value, while ACM and Freebase achieve the best performance. This indicates that it is important to utilize the cross-view mechanism to additionally learn the similarity

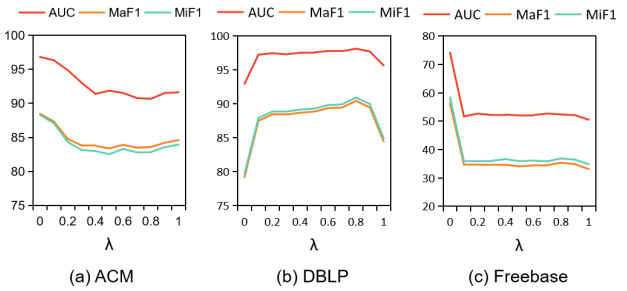

FIGURE 9. Analysis of hyperparameters.

between different meta-paths for datasets where the number of different meta-paths varies substantially (i.e., imbalanced). Since the majority of real-world datasets are imbalanced, the above findings illustrate the significance of utilizing both the cross-schema and cross-view parts for multi-scale learning.

VI. CONCLUSION

In this paper, we propose a novel self-supervised heterogeneous graph representation learning method called SNMH. To capture rich self-supervised signals, we conduct dual-schema view generation to obtain meta-path based views and relation type based 1-hop views, which represent the global and local information in a heterogeneous graph, respectively. Based on the Siamese network, SNMH implements a multi-scale bootstrapping contrastive learning mechanism to learn node representations in heterogeneous graphs from cross-schema and cross-view aspects. Our method does not require any negative samples, thus reducing time and space costs. Experimental results show that our method is superior to other methods. In the future, we will continue to research more efficient methods with lower time and space complexity.

ACKNOWLEDGMENT

The authors are grateful to the anonymous referee, who made valuable suggestions to help improve the article.

REFERENCES

[1] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a 'siamese' time delay neural network," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 6, 1993, pp. 669–688.
[2] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 539–546.
[3] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," Oct. 2017, *arXiv:1710.10903*.
[4] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[5] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks," in *Social Network Data Analytics*. 2011, pp. 115–148.
[6] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, *arXiv:1706.02263*.
[7] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[8] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," 2018, *arXiv:1806.01973*.

[9] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016, *arXiv:1611.07308*.
[10] 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. Eur. Semantic Web Conf.*, Jun. 2018, pp. 593–607.
[11] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 5165–5175.
[12] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 3844–3852.
[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[14] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.
[15] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.
[16] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Chang, "Network representation learning with rich text information," in *Proc. 24th Int. joint Conf. Artif. Intell.*, 2015, pp. 1–7.
[17] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Collective classification via discriminative matrix factorization on sparsely labeled networks," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 1563–1572.
[18] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.
[19] P. Velickovic, W. Fedus, W. L. Hamilton, P. Lio, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," 2018, *arXiv:1809.10341*.
[20] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 4116–4126.
[21] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Deep graph contrastive representation learning," 2020, *arXiv:2006.04131*.
[22] N. Jovanović, Z. Meng, L. Faber, and R. Wattenhofer, "Towards robust graph contrastive learning," 2021, *arXiv:2102.13085*.
[23] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *Proc. Web Conf.*, Apr. 2021, pp. 2069–2080.
[24] 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, Aug. 2011.
[25] C. Park, D. Kim, J. Han, and H. Yu, "Unsupervised attributed multiplex network embedding," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, Apr. 2020, pp. 5371–5378.
[26] Y. Ren, B. Liu, C. Huang, P. Dai, L. Bo, and J. Zhang, "Heterogeneous deep graph infomax," 2019, *arXiv:1911.08538*.
[27] X. Wang, N. Liu, H. Han, and C. Shi, "Self-supervised heterogeneous graph neural network with co-contrastive learning," 2021, *arXiv:2105.09111*.
[28] S. Thakoor, C. Tallec, M. G. Azar, M. Azabou, E. L. Dyer, R. Munos, P. Veličković, and M. Valko, "Large-scale representation learning on graphs via bootstrapping," 2021, *arXiv:2102.06514*.
[29] M. Jin, Y. Zheng, Y.-F. Li, C. Gong, C. Zhou, and S. Pan, "Multi-scale contrastive Siamese networks for self-supervised graph representation learning," 2021, *arXiv:2105.05682*.
[30] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised learning," 2020, *arXiv:2006.07733*.
[31] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *Proc. World Wide Web Conf.*, May 2019, pp. 2022–2032.
[32] J. Zhao, X. Wang, C. Shi, Z. Liu, and Y. Ye, "Network schema preserved heterogeneous information network embedding," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, 2020, pp. 1–7.
[33] X. Fu, J. Zhang, Z. Meng, and I. King, "MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proc. Web Conf.*, 2020, pp. 2331–2341.
[34] X. Li, D. Ding, B. Kao, Y. Sun, and N. Mamoulis, "Leveraging meta-path contexts for classification in heterogeneous information networks," in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*, Apr. 2021, pp. 912–923, doi: 10.1109/ICDE51399.2021.00084.
[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[36] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019, doi: 10.1109/TKDE.2018.2833443.
[37] 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.
[38] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

ZIJUN CHEN was born in Zibo, Shandong, China, in 2000. She is currently pursuing the B.S. degree in computer science with Shandong University, China. Her research interests include machine learning and graph-related learning.

LIHUI LUO was born in Yongcheng, Henan, China, in 2000. He is currently pursuing the B.S. degree in software engineering with Shandong University, China. His research interests include machine learning and recommender systems.

XUNKAI LI was born in Harbin, Heilongjiang, China, in 2000. He is currently pursuing the B.S. degree in computer science with Shandong University, China. His research interests include machine learning, graph-related learning, and intelligent information processing.

BIN JIANG received the D.S. degree from the University of Chinese Academy of Sciences. Since May 2005, he has been with Shandong University, Weihai, where he is an Associate Professor with the Department of Computer Science. His current research interests include astronomical data mining and machine learning algorithms.

QIANG GUO (Member, IEEE) received the D.S. degree from Shanghai University. Since 2012, he has been with the Shandong University of Finance and Economics, Jinan, where he is currently a Professor with the School of Computer Science and Technology. His current research interests include machine learning algorithms and data analysis.

CHUNPENG WANG received the M.S. degree from Shandong University. Since September 2011, he has been with Shandong University, Weihai, where he is currently a Research Associate with the Department of Computer Science. His current research interests include machine learning algorithms and human–computer interaction.