IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 February 2024, accepted 12 February 2024, date of publication 14 February 2024, date of current version 23 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3366231

RESEARCH ARTICLE

HGNN-QSSA: Heterogeneous Graph Neural Networks With Quantitative Sampling and Structure-Aware Attention

QIN ZHAO[1,2], (Senior Member, IEEE), YARU MIAO[1], DONGDONG AN[1], JIE LIAN[1], AND MAOZHEN LI[3]

[1]Shanghai Engineering Research Center of Intelligent Education and Big Data, Shanghai Normal University, Shanghai 200234, China
[2]Key Laboratory of Embedded Systems and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, China
[3]Department of Electronic and Computer Engineering, Brunel University London, UB8 3PH London, U.K.

Corresponding author: Jie Lian (lianjie@shnu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB4501704; in part by the National Natural Science Foundation of China under Grant 62302308, Grant U2142206, Grant 62372300, and Grant 61702333; in part by the Shanghai Sailing Program under Grant 21YF1432900; in part by the Shanghai Engineering Research Center of Intelligent Education and Big Data; and in part by the Research Base of Online Education for Shanghai Middle and Primary Schools.

ABSTRACT Heterogeneous information networks provide abundant structural and semantic information. Two main strategies for leveraging this data include meta-path-based and meta-path-free methods. The effectiveness of the former heavily depends on the quality of manually defined meta-paths, which may lead to the instability of the model. However, the existing meta-path-free methods lack of neighbor screening during aggregating, and there is also an overemphasis on attribute information. To address these issues, we propose the Heterogeneous Graph Neural Network model by incorporating Quantitative Sampling and Structure-aware Attention. We introduce a Quantitative Sampling Module that calculates the similarity between neighbors of the target nodes and target nodes, enabling us to select the top k nodes with the strongest relevance to the target node based on this measure, and incorporate a Structure-aware Attention Module during the aggregation of neighbor information. This module combines both structural and attribute information to aggregate the neighbor information effectively. By implementing these improvements, our proposed model exhibits superior performance compared to several state-of-the-art methods on two real-world datasets.

INDEX TERMS Heterogeneous information network, community detection, quantitative sampling, structure-aware attention.

I. INTRODUCTION

Community detection is a fundamental and crucial technique in complex network analysis, used to reveal the relational structure among network nodes. This process involves evaluating the connections between nodes in a social network. This evaluation results in the categorization of the social network. The categorization is based on these metrics, dividing the social network into different community structures. Community detection finds application in various domains, including social recommendation algorithms [1] that rely on community detection. It is worth noting that entities within the same community tend to have higher similarity and connectivity. Conversely, those between different communities typically exhibit weaker connections. Numerous algorithms have been proposed for community detection [2], [3], [4], [5], with most of them focusing on homogeneous graphs which are graphs composed of nodes and edges of the same type. However, these methods do not yield satisfactory results when applied to community detection tasks in heterogeneous information networks, which consist of nodes or edges of multiple types, while heterogeneous networks are prevalent in real-world scenarios. For instance, Fig. 1 illustrates a heterogeneous academic network featuring four types of

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

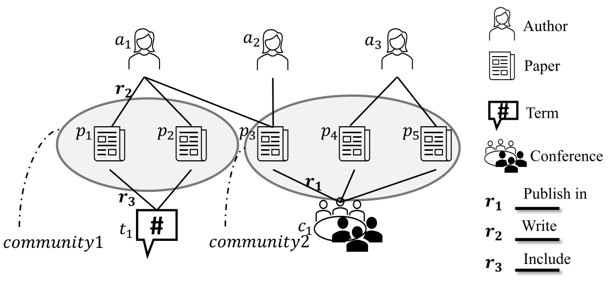


FIGURE 1. **Example for heterogeneous academic network.**

nodes: *author*, *paper*, *term*, and *conference*, along with three distinct types of edges, which are *write*, *include*, *publish in*.

Heterogeneous information networks offer a wealth of information, and by taking into account multiple types of edges and nodes, we can enhance the semantic and structural understanding of the network. When it comes to pattern-rich heterogeneous information networks, incorporating high-order relationships and multiple kinds of nodes poses an additional challenge for researchers, but considering high-order relationships can uncover implicit information. For instance, in Fig. 1, paper p_1 and p_2 are not directly connected. If we only consider the node type *paper*, it becomes difficult to classify them as belonging to the same community. However, by considering the high-order relationships of the nodes, it becomes possible to classify p_1 and p_2 into the same community. This is because both papers are authored by author a_1 and contain the same term t_1. Taking these high-order relationships into account allows for a more accurate classification of nodes within the network.

To address the limitations of community detection on homogeneous graphs, several approaches based on heterogeneous graph neural networks are being developed. These approaches include the heterogeneous graph attention network (HAN) [6], which applies attention mechanisms at different levels to capture hierarchical structure and key features in data. Layer-aware heterogeneous graph information network embedding (HAHE) [7], which leverages attention mechanisms to embed heterogeneous graph structures, capturing diverse node relationships and enhancing representation learning and metapath aggregated graph neural network (MAGNN) [8]. They use artificially defined meta-paths to reveal high-order potential relationships among nodes. These models construct homogeneous graphs through meta-paths and learn node representations using traditional GNN (Graph Neural Network) models. HMSN [9] extracts potential relationships between similar node types by utilizing meta-paths and similarity. Although these meta-path-based methods achieve some level of success, it is not possible to explore all meta-paths, and the quality of the predefined meta-paths significantly influences the algorithm's performance. Therefore, researchers explore meta-path-free methods which aggregate neighbor information by projecting different types of neighbor nodes into the same vector space, such as HGT [10], Simple-HGN [11], HetGNN [12], SeHGNN [13]. They employ attention mechanism to determine the importance of different neighbors. HSim [14] considers both the attributes and structure between nodes to more accurately compute the similarity between two nodes. SemAttNet [15] proposes a multimodal attention-based fusion module based on semantic perception to integrate features. Pa-mvsnet [16] introduces self-attention layers for hierarchical features extraction, which is able to capture multi-scale matching clues for the subsequent depth inference task. Hence, the quality of the selected neighbors directly affects the algorithm's performance. This is because the information from neighboring nodes is used to compute and predict the attributes or state of the target node. If the selected neighboring nodes are of high quality, the prediction results of the algorithm are more likely to be accurate. Conversely, if the selected neighboring nodes are of low quality, the prediction results of the algorithm may be biased. However, many meta-path-free approaches select all neighbors within a neighborhood without considering their relevance to the target node. These methods typically select all neighboring nodes within a neighborhood, regardless of whether these nodes are relevant to the target node. Consequently, the target node may not be able to focus on more valuable neighbor information. While some methods consider selecting neighborhood nodes that contain more valuable information for aggregation, they overlook the structural correlation between nodes. The structural correlation between nodes refers to the fact that the attributes or state of a node may be related to the attributes or state of its neighboring nodes. These approaches primarily rely on attribute correlation between nodes to assign attention weights, without adequately exploring the effect of structural correlation on neighbor importance. This approach may limit the model's expressiveness and lead to oversmoothing after multiple iterations. Since attributes and structure represent different heterogeneous spaces, it is crucial to strike a balance between attribute and structural correlation when calculating attention values. In this way, we can get more accurately attributes of the target node, thereby improving the performance of the algorithm.

To better address the first problem mentioned above, which involves utilizing all nodes in the neighborhood without selection, we propose the Quantitative Sampling Module (QSM). The goal of this module is to select the top k nodes that are most relevant to the target node. First, random walk is performed on the target node's subgraph to determine the sampling frequency for neighboring nodes. The structural relevance score between the target node and its neighboring nodes is calculated using this sample frequency. The feature relevance score between the target node and its neighboring nodes is then calculated as the cosine similarity between the target node and its neighboring nodes. Zhao et al. [17] assess node similarity by determining whether they contain the same keywords. They introduce semantic similarity when measuring node similarity in [18]. The structural relevance score and featural relevance score are then multiplied together

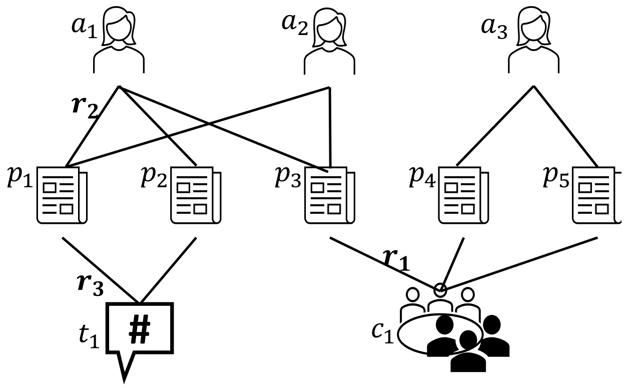


FIGURE 2. Example for heterogeneous academic network.

to rank the nodes, allowing us to select the top k nodes with the highest relevance to the target node for information aggregation. This approach enables us to focus more attention on nodes that are most relevant to the target node, leading to more efficient and accurate capture of the target node's representation. Regarding the second issue, previous studies have proposed incorporating structural information into the calculation of attention coefficients. However, these methods designed for homogeneous networks cannot be directly applied to heterogeneous information networks. Therefore, in this paper, we introduce the Structure-aware Attention Module (SAM). In social networks, social influence often leads individuals and their friends to exhibit similar behaviors [19]. Taking inspiration from this, we assume the existence of social influence among homogeneous nodes that share similar network topology. For example, in Fig. 2, authors a_1 and a_2 co-authored paper p_1 and p_3, possibly indicating a social relationship such as a teacher-student or colleague relationship. Building upon this idea, we calculate the topological correlation between nodes to determine the structural attention scores. Additionally, to prevent oversmoothing caused by layer stacking, we incorporate a residual structure into our model.

The major contributions of this paper are summarized as follows:

- We propose the Quantitative Sampling Module (QSM) to better select neighboring nodes with a high relevance to the target node. The module is constructed by calculating the strctural and attribute relavance scores between target node and its neighboring nodes.
- We present the Structure-aware Attention Module (SAM), which combines both structural and attribute information to determine the contribution of nodes in the neighborhood to the target node. To address the issue of oversmoothing caused by layer stacking, we adopt a residual structure in the final representation layer. This helps mitigate the potential negative impact of excessive layer stacking and improves the overall performance of the model.
- To validate the rationality and effectiveness of the proposed model, we conduct comparative experiments on two real-world datasets. The experimental results demonstrate the superiority of the algorithm proposed in this paper. The proposed model outperforms other algorithms in terms of Micro-F1 and Macro-F1. Furthermore, we perform ablation experiments to verify the effectiveness of the two modules within the proposed algorithm. The parameter analysis is conducted to investigate the impact of the hyperparameters on the algorithm performance.

The rest of the paper is organized as follows. Section II describes some related work about community detection. Section III introduces the symbols and concepts used in this paper. In Section IV, we propose our model HGNN-QSSA. The experimental results and the influencing factors are discussed in Section V. Finally, conclusions are described in Section VI.

II. RELATED WORK

In this section, we will review the relevant work that primarily focuses on community detection tasks: 1) Traditional community detection algorithms; 2) GNNs for community detection; and 3) HGNNs for community detection.

A. TRADITIONAL COMMUNITY DETECTION ALGORITHMS

Due to the importance of community structure in social networks, various algorithms for community detection are developed [20], [21], [22], [23], [24], including both traditional and deep learning approaches. We direct our attention towards the algorithms associated with graph partitioning and label propagation [25] from the traditional approach. The problem of graph partitioning involves dividing nodes into groups of a predefined size, aiming to minimize the number of edges between the groups. Graph partitioning is the distribution of data to different nodes in a cluster in distributed computing, and the design of many serial algorithms. There are two ways to partition a graph: by nodes and by edges [26], each with its own advantages and disadvantages. Most variants of the graph partitioning problem are known to be NP-hard [27]. A representative method for solving this problem is the Kernighan-Lin algorithm [28], which optimizes a benefit function called Q [29]. The algorithm initially partitions the social network graph into several communities of a predetermined size at random and evaluates the gain in Q by changing the composition of nodes within the communities. The Label Propagation Algorithm (LPA) is a local (bottom-up) partitioning algorithm inspired by the spread of epidemics [30]. In this algorithm, the input graph is first preprocessed, and during the initialization phase, each node is assigned a usually unique label. Iterative label propagation follows, where in each iteration, the label of each node is updated. Variants of the algorithm allow more than one label to be assigned to each node, enabling the definition of overlapping communities. These traditional algorithms provide valuable insights and techniques for community detection tasks in social networks. However, traditional methods have certain limitations in terms of performance, such as the label propagation algorithm having a degree of randomness.

B. GNNS FOR COMMUNITY DETECTION

Several deep learning-based algorithms have been investigated for community detection. Among them, GNN [31] is a widely recognized algorithm that exhibits strong performance. GCN [32] also used to detect community, which efficiently captures intricate features from network topology and node attributes by employing a series of convolutional operations, similar to how CNN (Convolutional Neural Network) [33] operates. Thus, some methods employ it to represent the node information [34]. Zhao et al. employ Graph Attention Network to aggregate neighborhood information by computing attention coefficients for neighboring nodes [35], [36], which allowing the algorithm selectively emphasize the most relevant nodes during information aggregation. GraphSAGE [37] performs a neural network-based aggregator on a fixed-size neighborhood of nodes. It is an inductive representation learning type which can gengrate node representations effectively. Zhang et al. [38] propose randomly permuting training data as a simple data augmentation strategy, which not only enhances the model's robustness to variations in the input data but also aids in preventing overfitting. MRFasGCN [39] is an extension to GCN, which leverages Markov random walks to enhance the performance of graph neural networks in node classification tasks.

These deep learning-based algorithms provide promising approaches to community detection by leveraging the power of neural networks and effectively incorporating network structure and node attributes into the learning process.

C. HGNNS FOR COMMUNITY DETECTION

Given the suboptimal performance of GNNs on heterogeneous networks, significant efforts have been devoted to exploring HGNNs. HAN [6] utilizes meta-paths to capture high-order information of nodes. It computes attention scores on subgraphs generated based on these meta-paths and further incorporates inter-type attention for information aggregation. Another approach, HAHE [7] computes attention weights for meta-paths and subsequently calculates node-level attention scores. MAGNN [8] builds upon HAN by enhancing the information of intermediate nodes during meta-path aggregation. HetGNN [12] proposes a technique that utilizes random walk with restart to sample nodes and aggregates information from similar neighbors. It employs an attention mechanism between different types of neighbors for information aggregation. HetSANN [40] formulates the transformations between diverse nodes by projecting them into a lower-dimensional entity space. Subsequently, it employs a graph neural network to aggregate multi-relational information within the projected neighborhoods, leveraging attention mechanisms.

However, the aforementioned approaches either rely solely on meta-paths for information aggregation, potentially underutilizing the network topology, or they overlook the importance of structural attention and fail to fully exploit the structural information present in the network.

III. NOTATIONS AND DEFINITIONS

In this section, we will introduce the notations used in this paper and subsequently provide the problem definition addressed in this study.

A. NOTATIONS

The heterogeneous information network is described as $G = (V, E, A, R)$ with a node-type mapping function $\phi : V \rightarrow A$, and an edge-type mapping function $\psi : E \rightarrow R$, where V and E represent the set of nodes and edges, respectively, and A and R are set of node types and edge types, respectively. The heterogeneous information network should have different types of nodes or edges such that $|A| + |R| > 2$, and we use ϕ^i to denote the type of node i.

B. DEFINITIONS

A heterogeneous network is a complex network of several relationships, which contains multiple types of nodes and edges. These nodes represent different entities in the specific domain, such as *paper*, *author*, *conference*, *term*, etc., while edges represent the associative relationships between different entities, such as *paper citations*, *author collaborations*, etc. Heterogeneous academic networks are rich in information, in which semantic and structural information can be enhanced by considering different types of edges and nodes.

Definition 1 (Primary Type): Due to the presence of multiple node types, it is necessary to designate one node type as the primary type for this study. In this paper, the *paper* node is selected as the primary type. However, it is important to note that theoretically any node type could be chosen as the primary type.

Definition 2 (Target node): To generate a subgraph g_i for each target node i, we select the node from the primary type as the target node and then choose its n-hop neighbors as the subgraph.

Definition 3 (Node Embedding): Given the heterogeneous graph $G = (V, E, A, R)$ and the primary node type, the node embedding is obtained by aggregating the neighbor information of the target node and then projecting the target node into a low-dimensional space. Our primary objective is to learn a mapping function $f : V \rightarrow R_d$ that projects each primary type node to R_d in which $d \ll$ the number of target nodes.

IV. METHODOLOGY

In this section, we formally introduce the HGNN-QSSA model to address the two issues described in the previous section. Aggregating all neighbor information without selection may result in the aggregation of less valuable neighbor information, while calculating attention coefficients based solely on attribute information can limit the model's expressiveness. Hence, we proposed our model to address the issues mentioned above, which consists of two main components: 1) Quantitative Sampling Module. This module selects the k neighbors most strongly related to the target node for information aggregation, allowing for the aggregation of more valuable neighbor node information. 2) Structure-aware Attention Module. This module simultaneously considers attribute similarity and structural similarity, computing their joint attention coefficients for final information aggregation, thus achieving a balance between attributes and structure. The framework of HGNN-QSSA is illustrated in Fig. 3.

A. QUANTITATIVE SAMPLING MODULE

Quantitative Sampling Module (QSM) involves conducting a random walk within the subgraph of the target node to obtain sampling probabilities for its neighboring nodes. The sampling probabilities are employed as measures of structural relevance between the target node and its adjacent nodes. In addition, they assess the similarity in attributes between the target node and its neighbors. The comprehensive relevance of nodes within the sensory domain to the target node is calculated by the product of the structural relevance score and attribute similarity. Subsequently, top k neighbors of each type are selected.

We perform a random walk in subgraph g_i of the target node i, which can determine the structural correlation between the target node i and the neighboring nodes, based on the sampling frequency. The walker iteratively travels to each neighbor with a probability to edge weight. The iterative formula can be expressed as follows:

$$\overrightarrow{r_i} = \widetilde{\mathbf{W_i}}\overrightarrow{r_i} + \overrightarrow{e_i}, \tag{1}$$

where $\overrightarrow{r_i}$ represents the structural relevance scores (i.e., steady-state probabilities) of all the neighbors in the subgraph g_i with respect to target node i, and $\widetilde{\mathbf{W_i}}$ is derived as a transition probability matrix by normalizing the columns of the adjacency matrix $\mathbf{W_i}$, where $\widetilde{\mathbf{W_i}}[\mathbf{i}, \mathbf{j}]$ represents the probability of walking from j to i. Additionally $\overrightarrow{e_i}$ is an initial vector with its ith element is 1, and the rest of the elements are 0. Now that we have obtained the structural relevance scores, the next step is to calculate the attribute relevance score between the target node and its neighboring nodes in the perceptual field. However, before proceeding with this calculation, it is necessary to transfer the representations of different types of nodes to the same space. This ensures that the node representations are compatible and can be effectively compared and aggregated for further analysis.

$$h^{l+1}_{\phi^i,j} = \mathbf{w}^{l+1}_{\phi^i,\phi^j} h^l_j, \tag{2}$$

where h^l_j denotes the representation of node j at layer l. The transformation matrix $\mathbf{W^{l+1}_{\phi^i,\phi^j}}$ is used to transfer node j to the node representation space of node i. The representation of node j in the ϕ^i space is denoted as $h^{l+1}_{\phi^i,j}$. This transformation ensures that the representations of nodes from different types are aligned in the same feature space, facilitating subsequent attribute correlation calculation and information aggregation. Then we can calculate the cosine similarity between target node and its neighbors in the perceptual field, in which cosine similarity denotes the attribute correlation:

$$f^{\phi^i\phi^j}_{ij} = \cos(\theta) = \frac{h_{\phi^i,i} \cdot h_{\phi^i,j}}{||h_{\phi^i,i}|| \cdot ||h_{\phi^i,j}||}, \tag{3}$$

where $f^{\phi^i\phi^j}_{ij}$ denotes the cosine similarity between node i and node j, and $h_{\phi^i,i}$ and $h_{\phi^i,j}$ denote the feature representations of node i and node j in feature space node of i, respectively. The cosine similarity calculation measures the similarity between the attribute-level of two nodes, providing a quantitative measure of their similarity in the feature space. Then, the structural correlation and attribute correlation between the target node and its neighboring nodes have been obtained. The next step involves calculating the combined correlation between the target node and its neighboring nodes to determine which neighbors to sample. This process can be described as follows:

$$\mathrm{corr}^{\phi^i\phi^j}_{i,j} = f^{\phi^i\phi^j}_{ij} \cdot r^{\phi^i\phi^j}_{ij}, \tag{4}$$

where $corr^{\phi^i\phi^j}_{i,j}$ represents the combined correlation of node i and node j. The top-k neighbours of each type are selected as the neighbours of the target node based on the ranking of $corr^{\phi^i\phi^j}_{i,j}$. Fig. 4 displays this calculation process. Note that when k exceeds the total count of a particular type of neighbor within the neighborhood of the target node, we opt to fill it using the mean of representation of the existing neighbor. For example, consider a paper p_1, with two terms, but when k is set to 3. We initially select these two terms, and then calculate the average representation of these two terms as the representation for the third term, which is subsequently aggregated.

After that, we have successfully selected the set of neighboring nodes of the target node i, which will be utilized to aggregate information, and the set is denoted as A_i.

B. STRUCTURE-AWARE ATTENTION MODEL

In this section, we propose SAM to balance the relationship between attribute and structure when calculating the attention coefficient between nodes. SAM have two steps, including node-level attention and type-level attention. We employ a dual-level attention mechanism because heterogeneous networks encompass diverse node types and multiple relationships. The utilization of type-level attention serves to facilitate the model in distinguishing the relative importance of different node types, thereby enhancing the quality of information aggregation. Simultaneously, node-level attention helps the model in the selective aggregation of important neighboring nodes within the same node type, thus avoiding indiscriminate aggregation of all neighbors. Next, we will provide a detailed explanation of these two components.

1) NODE-LEVEL ATTENTION

Node-level attention allows us to learn the importance weights of neighbors belonging to the same type. We first

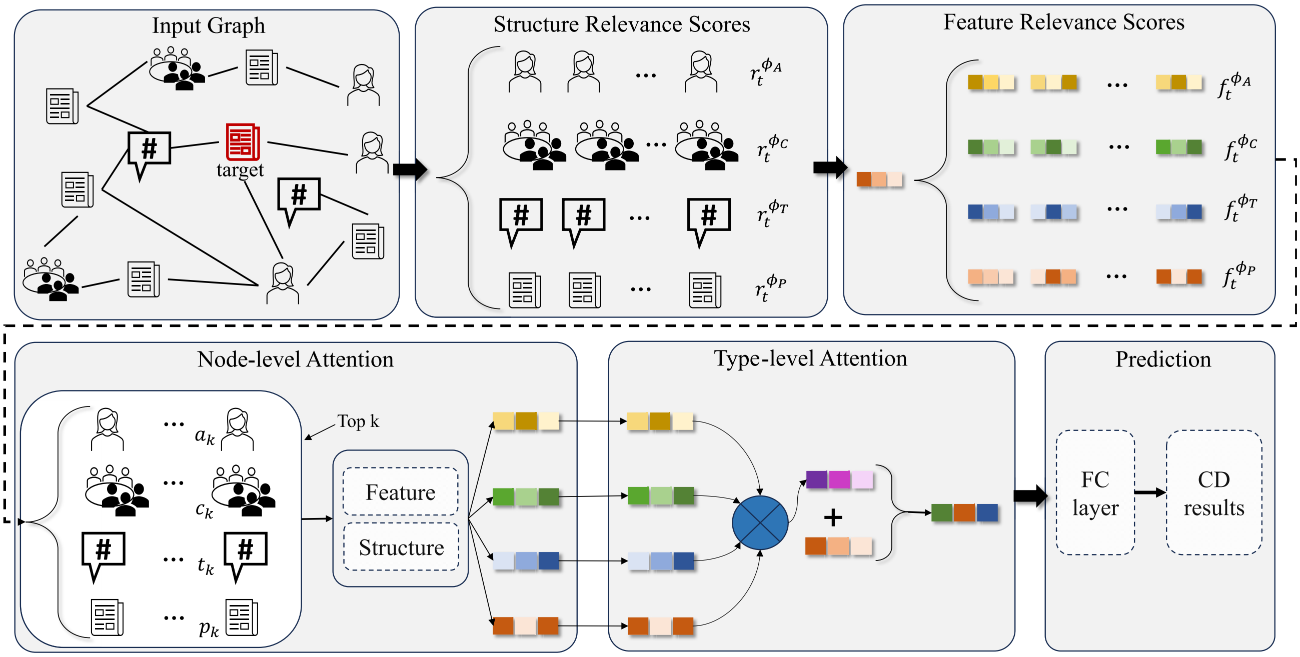


FIGURE 3. Overall architecture of the HGNN-QSSA model.

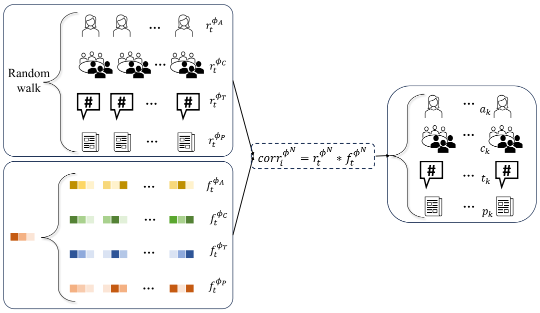


FIGURE 4. The top-*k* node selection process.

compute feature-based attention scores for the target node i and its neighboring nodes. For $j \in A_i$, of type ϕ^j, the attention coefficient e_{ij} can be calculated as

$$e_{ij}^{l+1} = leakyReLU\left(a_r^T [h_{\phi^i,i}^{l+1} || h_{\phi^j,j}^{l+1}]\right), \tag{5}$$

where e_{ij}^{l+1} projects the splice of node i and node j onto a real number. a_r^T is a trainable parameter and r represents the relationship of edges. By normalizing the e_{ij}^{l+1},the feature-based attention coefficient a_{ij}^{l+1} can be obtained by softmax normalization of e_{ij}^{l+1}:

$$\alpha_{ij}^{l+1} = \frac{\exp(e_{ij}^{l+1})}{\sum_{v \in \phi^i} \exp(e_{iv}^{l+1})}. \tag{6}$$

However, this attention score only considers feature information and does not incorporate structural information. Based on the structural relevance score $r_{ij}^{\phi^i\phi^j}$ and feature relevance score $f_{ij}^{\phi^i\phi^j}$ of target node and its neighboring nodes calculated in the previous section, we calculate the structural similarity S_{ij} of node i and node j. First, we need to calculate the structural similarity in the neighborhood of node i and node j for each type of neighbor:

$$S_{ij}^{\phi_p} = \frac{\sum_{u \in N(v_j)} \left(f_{ju}^{\phi^j\phi^u} \cdot r_{ju}^{\phi^j\phi^u}\right)}{\sum_{v \in N(v_i)} \left(f_{iv}^{\phi^i\phi^v} \cdot r_{iv}^{\phi^i\phi^v}\right)}, \tag{7}$$

where $S_{ij}^{\phi_p}$ denotes the structural similarity between node i and node j in type p. u and v represent the neighbors of node j and node i of type p, respectively. The final structural correlation between node i and node j can be represented as:

$$S_{ij} = \sum_{p \in A} S_{ij}^{\phi_p}. \tag{8}$$

Now that we have obtained the structural similarity between nodes, the next step is to calculate the structure-aware attention coefficient. The attention coefficients of neighbors from different types compared to the target node are calculated using $r_{ij}^{\phi^i\phi^j}$, and the attention coefficients of neighbors from same type compared to the target node are calculated using S_{ij}:

$$\beta_{ij}^{l+1} = \frac{\exp\left(r_{ij}^{\phi_i\phi^j}\right)}{\sum_{v \in A_i^{\phi^v}} \exp\left(r_{iv}^{\phi_i\phi^v}\right)}, \tag{9}$$

$$\beta_{ij}^{l+1} = \frac{\exp(S_{ij})}{\sum_{v \in A_i} \exp(S_{iv})}. \tag{10}$$

So far, we have obtained the feature-based and structure-based attention coefficients for node i and node j respectively. Now, we need to combine these two attention coefficients to obtain the node-level importance coefficients γ_{ij}^{l+1}, which are combined in the following way:

$$\gamma_{ij}^{l+1} = t\alpha_{ij}^{l+1} + (1-t)\beta_{ij}^{l+1}, \tag{11}$$

where t is a hyperparameter to control the proportion of feature and structural importance. According to the importance coefficient obtained γ_{ij}^{l+1} can be used to aggregate the neighbors of node A_i in the same type of neighbourhood information:

$$\mathbf{H}_{\mathbf{i},\phi_\mathbf{p}}^{\mathbf{l+1}} = \sigma\left(\sum_{j\in A_i^{\phi_p}} \gamma_{ij}^{l+1} \cdot h_{\phi^{i}j}^{l+1}\right), \tag{12}$$

where $A_i{}^{\phi_p}$ denotes p type nodes in neighborhood of node i, and $\mathbf{H}_{i,\phi_p}^{l+1}$ denotes the aggregation of all neighbors of the target node i in the p-type. In this step, we obtain the representations of all nodes of all types in the neighborhood of node i denoted as $\{\mathbf{H}_{\mathbf{i},\phi_1}^{\mathbf{l+1}}, \mathbf{H}_{\mathbf{i},\phi_2}^{\mathbf{l+1}}, \ldots, \mathbf{H}_{\mathbf{i},\phi_\mathbf{N}}^{\mathbf{l+1}}\}$ by combining information of all neighbors from different types in the neighborhood of node i.

2) TYPE-LEVEL ATTENTION

In the previous step, we calculate the importance values between neighbors of node i and its neighbors in neighborhood and aggregate them between neighbors of the same type. Now, we move on to calculate the type-level attention values, which is based on the node representation of all types in the neighborhood of node i obtained in the previous step. This process involves capturing the significance of each neighbor type in contributing to the overall representation of node i. We proceed with the following step to obtain the type-level attention values:

$$\omega_i^{\phi_p} = \frac{\exp\left(Sigmod\left(\alpha\left[\mathbf{WH}_{\mathbf{i},\phi_\mathbf{i}}^{\mathbf{l+1}}, \mathbf{WH}_{\mathbf{i},\phi_\mathbf{p}}^{\mathbf{l+1}}\right]\right)\right)}{\sum_{P=1}^{P} \exp\left(Sigmod\left(\alpha\left[\mathbf{WH}_{\mathbf{i},\phi_\mathbf{i}}^{\mathbf{l+1}}, \mathbf{WH}_{\mathbf{i},\phi_\mathbf{p}}^{\mathbf{l+1}}\right]\right)\right)}, \tag{13}$$

where $\omega_i^{\phi_p}$ represents the importance of p type neighbors. After obtaining the attention coefficients between node types, the next crucial step is type-level neighbor aggregation. To address the issue of gradient disappearance as the network layers deepen, we introduce a residual unit in this process. The residual unit allows us to preseave importance information from the previous layers while aggregating information from type-level neighbors. The representation of node i after type-level neighbor aggregation can be described as follow:

$$h_i^{l+1} = \sum_{P=1}^{P} \omega_i^{\phi_p} \cdot \mathbf{H}_{\mathbf{i},\phi_\mathbf{P}}^{\mathbf{l+1}} + h_i^l, \tag{14}$$

where h_j represents the final representation of node i which encapsulates both the information from the node-level and type-level neighbors. Finally, we connect all the primary type nodes' final representations with a fully connected layer to achieve the community detection task.

The incorporation of the residual unit enhances the model's ability to learn complex hierarchical features from heterogeneous neighborhoods and helps capture more meaningful patterns in the data. By considering both node-level and type-level attention mechanisms, and effectively addressing the vanishing gradient problem, the proposed HGNN-QSSA framework exhibits improved performance within the heterogeneous information networks.

TABLE 1. The statistics of the public datasets.

Datasets	Nodes	Edges
DBLP	#author(A): 4057	# A-P: 19645
	#paper(P): 14328	# P-T: 85810
	#term(T): 7723	# P-C: 14328
	#conferences(C): 20	
ACM	#paper(P): 4019	# P-P: 9615
	#author(A): 7167	# P-A: 13407
	#subject(S): 60	# P-S: 4019

V. EXPERIMENT

A. DATASETS

We utilized two real-world heterogeneous academic network datasets, namely DBLP and ACM. Next, we will introduce the detailed information about these two datasets.

1) DBLP

The dataset utilized in our study is a subset extracted from the DBLP database, consisting of a diverse set of academic resources. Specifically, it includes 14,328 research papers, contributed by 4,057 authors, across 20 different academic conferences. The dataset also encompasses 8,789 unique terms, each representing essential concepts discussed within the papers [6].

2) ACM

This dataset is a subset extracted from ACM [6], consisting of a diverse collection of academic resources. It comprises 4,019 research papers, authored by 7,167 scholars, and covers 60 distinct subjects of study. To provide a comprehensive representation of the papers and authors, various attributes have been incorporated into the dataset.

Table 1 demonstrates the number of nodes, edges, node types and edge types of the two datasets.

B. BASELINES

We compared some classical methods used in graph neural networks, as well as advanced graph neural network methods designed for heterogeneous graphs. Below is a brief introduction to these methods.

1) GRAPHSAGE [37]

GraphSAGE is an algorithm for inductive node embedding in graphs. It addresses the problem of learning embeddings for nodes that are not seen during training, making it highly scalable. GraphSAGE samples and aggregates features from the neighborhood of each node to learn its representation. This approach enables the algorithm to capture both structural and semantic information, making it effective for various graph-based tasks.

2) GCN [32]

GCN is a revolutionary neural network architecture for learning node embeddings in graphs. It leverages graph convolutional layers to efficiently capture and propagate information through a graph's neighborhood. GCN has demonstrated remarkable performance in tasks such as node classification and link prediction by effectively modeling node relationships and graph structures.

3) GAT [41]

GAT is an innovative neural network architecture designed for graph-based learning tasks. It employs attention mechanisms to adaptively weigh the importance of neighboring nodes when aggregating information. GAT's ability to capture fine-grained dependencies in graphs and its self-attention mechanism make it a powerful tool for various applications, including node classification and graph classification.

4) METAPATH2VEC [42]

metapath2vec is a pioneering algorithm for learning embeddings in heterogeneous information networks. It introduces the concept of metapaths, which are paths composed of multiple node types, to capture both structural and semantic information. By leveraging metapaths, metapath2vec can generate rich and context-aware embeddings for various types of nodes in heterogeneous networks.

5) HAN [6]

HAN is a cutting-edge model designed for heterogeneous graph-based learning. It combines both node-level and metapath-level attention mechanisms to effectively capture local and global information in heterogeneous graphs. HAN has shown superior performance in various applications, including node classification, due to its ability to handle the complexity and diversity of heterogeneous graph data.

6) MAGNN [8]

MAGNN consists of three primary components: Node Content Transformation, Intra-Metapath Aggregation, and Inter-Metapath Aggregation. To elaborate, MAGNN initiates by mapping the attribute information of nodes from various types, which may possess varying dimensions, into a shared latent vector space via type-specific linear transformations. Following this, it employs metapath-specific aggregation methods with attention mechanisms for each metapath. During the intra-metapath aggregation phase, each target node extracts and amalgamates information from metapath instances connecting it to its metapath-based neighbors. Through this process, MAGNN effectively captures both the structural and semantic information present in the heterogeneous graph, encompassing details from neighboring nodes and the metapath-based relationships that connect them.

After the intra-aggregation, MAGNN further performs inter-metapath aggregation using attention mechanisms to fuse latent vectors obtained from multiple metapaths into the final node embeddings. It introduces a novel metapath aggregation graph neural network for heterogeneous graph embeddings.

7) HETGNN [12]

HetGNN addresses challenges in heterogeneous networks: how to sample strongly correlated neighboring nodes in a heterogeneous graph, how to design encoders for heterogeneous node content to address consistency issues, and how to consider the influence of nodes of different types during neighbor information aggregation. These challenges are tackled in three parts. First, it introduces a random walk method with a restart mechanism to effectively select relevant nodes in the heterogeneous graph. Second, to resolve the consistency issue arising from nodes with different content in a heterogeneous graph, HetGNN designs various encoding approaches for nodes of different types based on their attribute information. Finally, during the process of neighbor information aggregation, HetGNN employs Bi-LSTM for nodes of the same type and introduces an attention mechanism to jointly learn the impact of different types of neighbors. These strategies collectively enable HetGNN to effectively address the complexity of heterogeneous graph data, enhancing its performance and applicability.

8) SAHNE [43]

SAHNE designs a structure-aware heterogeneous network embedding model to simultaneously detect each node's community and organizational distribution and learn the embeddings of nodes, communities, and organizations.

C. EVALUATION

Our ultimate goal is to carry out community division. Here, we use $Micro - F1$ and $Macro - F1$ as evaluation metrics to synthesize the performance of the proposed method and the method mentioned in baseline. These two metrics can be formulated as:

$$\begin{aligned}
\mathrm{Precision}_{\mathrm{micro}} &= \frac{\sum_{i=1}^{n} \mathrm{TP}_i}{\sum_{i=1}^{n} \mathrm{TP}_i + \sum_{i=1}^{n} \mathrm{FP}_i}, \\
\mathrm{Recall}_{\mathrm{micro}} &= \frac{\sum_{i=1}^{n} \mathrm{TP}_i}{\sum_{i=1}^{n} \mathrm{TP}_i + \sum_{i=1}^{n} \mathrm{FN}_i}, \\
F1_{\mathrm{micro}} &= 2 \cdot \frac{\mathrm{Precision}_{\mathrm{micro}} \cdot \mathrm{Recall}_{\mathrm{micro}}}{\mathrm{Precision}_{\mathrm{micro}} + \mathrm{Recall}_{\mathrm{micro}}}. \qquad (15) \\
\mathrm{Precision}_{\mathrm{macro}} &= \frac{\sum_{i=1}^{n} \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i}}{n}, \\
\mathrm{Recall}_{\mathrm{macro}} &= \frac{\sum_{i=1}^{n} \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}}{n},
\end{aligned}$$

$$F1_{\text{macro}} = 2 \cdot \frac{\text{Precision}_{\text{macro}} \cdot \text{Recall}_{\text{macro}}}{\text{Precision}_{\text{macro}} + \text{Recall}_{\text{macro}}}, \tag{16}$$

where TP (True Positives) are the correctly predicted positive values, and FP (False Positives) are the negative values incorrectly predicted as positive. FN (False Negatives) are the positive values incorrectly predicted as negative. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall is the ratio of correctly predicted positive observations to all observations in actual class. $F1$ score is the weighted average of Precision and Recall, also known as the harmonic mean of Precision and Recall.

D. EXPERIMENTAL SETTINGS

Among the eight algorithms mentioned, GraphSAGE, GCN, and GAT are algorithms designed for homogeneous graphs. In homogeneous networks, they utilize the network's topology and node attributes to detect community structures within the network. On the other hand, Metapath2vec, HAN, MAGNN, and HetGNN are algorithms developed for heterogeneous graphs. The first three models employ metapath-based methods, while HetGNN takes a unique approach by using a random walk strategy with restart to sample the neighborhood of target nodes, without utilizing metapaths.

This paper introduces an approach for heterogeneous neural networks, which is a community detection algorithm based on quantitative sampling and structure-aware attention. The method consists of two main steps: quantitative sampling and structure-aware based aggregation. The second part involves fusion mechanisms at both the node-level and type-level.

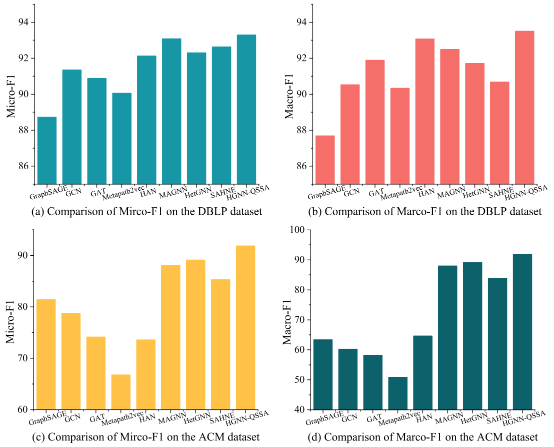
In our experimental setup, we established a learning rate of 0.01, a common choice that ensures stability during the model's training process. During the random walk process, our walk length is determined by the number of neighbors of the node, thereby enabling the adaptive acquisition of the walk sequence. We utilized the following versions of software packages: We employed PyTorch 1.2.0 as our deep learning framework, which offers robust tensor computation and deep neural network support. We used DGL 0.3.1 to handle data with graph structures. We utilized NetworkX 2.3 to create, manipulate, and study the structure and functions of complex networks. We applied scikit-learn 0.21.3 for training and evaluating machine learning models. We used NumPy 1.17.2 for efficient multi-dimensional array operations. Lastly, we used SciPy 1.3.1 for scientific and technical computing. These software packages and their versions were chosen based on the requirements of our experiments.

E. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we conduct a series of simulation experiments using two citation network datasets. The primary objective of these experiments is to comprehensively assess the performance of the HGNN-QSSA algorithm introduced in this paper. This assessment is achieved through a meticulous comparison of overall experimental outcomes and a detailed analysis of the algorithm's parameter settings.

TABLE 2. Comparison of different algorithms on two public networks.

Metrics	DBLP		ACM	
Network	Micro-F1	Macro-F1	Micro-F1	Macro-F1
GraphSAGE	88.73	87.69	81.43	63.39
GCN	91.36	90.53	78.79	60.24
GAT	90.88	91.89	74.16	58.21
Metapath2vec	90.06	90.34	66.8	50.86
HAN	92.13	93.07	73.61	64.66
MAGNN	93.09	92.49	88.11	88.01
HetGNN	92.31	91.71	89.14	89.17
SAHNE	92.64	90.68	85.32	83.91
HGNN-QSSA	**93.29**	**93.51**	**91.89**	**91.94**



(a) Comparison of Mirco-F1 on the DBLP dataset

(b) Comparison of Marco-F1 on the DBLP dataset

(c) Comparison of Mirco-F1 on the ACM dataset

(d) Comparison of Marco-F1 on the ACM dataset

FIGURE 5. Performance of HGNN-QSSA and its variants.

To shed light on the effectiveness of selecting strongly correlated neighbors, we carefully analyze and compare the experimental results of two critical components of our algorithm: Quantitative Sampling and Structure-aware Attention mechanisms. By doing so, we gain valuable insights into the impact of these elements on the algorithm's performance and validate the necessity of incorporating Structure-aware Attention in neural networks.

Our experiments and subsequent discussions not only provide empirical evidence of the algorithm's effectiveness but also contribute to a deeper understanding of the significance of structure-aware attention in enhancing the performance of graph-based neural networks.

1) ANALYSIS OF THE OVERALL EXPERIMENTAL RESULTS

First, we will conduct experiments and compare them with some advanced methods. As shown in Table 2, HGNN-QSSA performs best compared with current provides a comparison of Mirco-F1 and Macro-F1 on two real datasets. Meanwhile, for a clearer visualization of the experimental results, we have presented them in graphical form to better illustrate the effectiveness of our approach, as shown in the Fig. 5. Here are some key observations:

1. Apart from the Metapath2vec method, models designed based on heterogeneous networks demonstrate superior

overall performance compared to traditional models based on homogeneous networks, and this outcome is evident. The relatively lower performance of Metapath2vec compared to GCN, GAT, and others may be attributed to factors such as potential information loss, improper selection of metapaths during Metapath2vec sampling, and issues related to imbalanced node types.

2. The performance improvement of this model exhibits a notable discrepancy between the two datasets, with a more substantial performance boost observed on the ACM dataset compared to DBLP. This disparity arises from the inherent scarcity of node attribute information in the DBLP dataset. In the design process of the Quantitative Sampling Module, the selected nodes demonstrate a strong correlation with the target node. Therefore, through the development of more valuable neighbor selection and aggregation methods on the ACM dataset, the model's performance is effectively enhanced. This enhancement significantly bolsters its effectiveness, further confirming the efficiency of our QSM.

3. It is evident that MAGNN and HAN achieve suboptimal results on the DBLP dataset, while HetGNN attains suboptimal results on the ACM dataset. Notably, HetGNN is meta-path-free method. Both MAGNN and HAN are meta-path-based models and can directly provide task-specific meta-paths (e.g., paper classification). To elaborate, papers can be categorized into the relevant field using the *PCP* meta-path based on the submitted conference, which facilitate paper classification.

2) ABLATION EXPERIMENT

In order to assess the impact of various key components within the HGNN-QSSA model on overall performance, we conduct ablation studies by sequentially removing individual components on the DBLP and ACM datasets. The experimental results are presented in Fig. 6.

In the first step, we removed the Quantitative Sampling module (QSM) by aggregating neighborhood nodes indiscriminately instead of selecting the top k most relevant neighboring nodes. Denoted as "w/o QS," the results show a significant drop in model performance when this module is removed. In the second step, we removed the Structure-aware Attention module (SAM) by omitting the computation of structural similarity between nodes and solely using attribute similarity for aggregation through attention mechanisms. Denoted as "w/o SA," the experimental results demonstrate a more pronounced decrease in model performance, emphasizing the effectiveness of appropriately calculating structural similarity between nodes during the process of node information aggregation.

3) ANALYSIS OF PARAMETER

In this section we examined the sensitivity of hyperparameters in HGNN-QSSA on the DBLP and ACM datasets. Specifically, K represents the number of neighbors for quantitative sampling, and Embedding Size indicates the dimension of node embeddings.

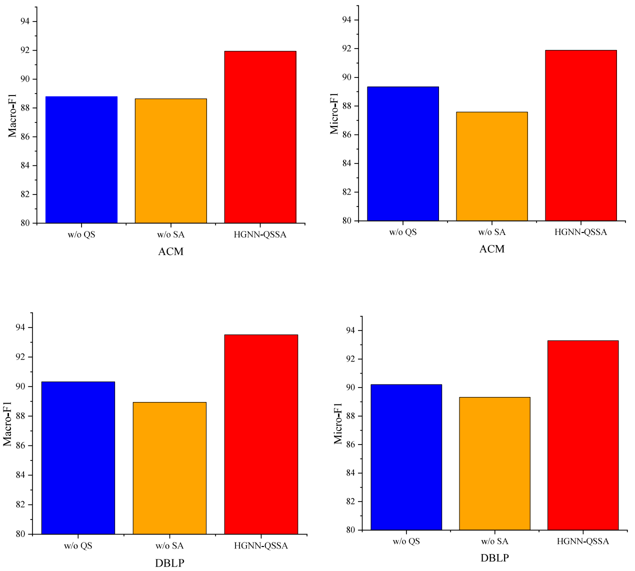


FIGURE 6. Performance of HGNN-QSSA and its variants.

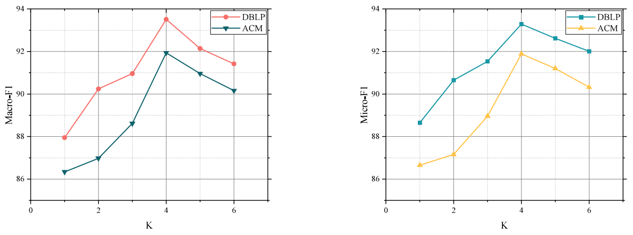


FIGURE 7. Parameters analysis over DBLP and ACM on sampling neighbors number K.

a: ANALYSIS OF K

We test the performance of HGNN-QSSA with different values of K. The range of K values tested is from 1 to 6. As shown in Fig. 7, it can be observed that as K increases, the model's expressiveness initially improves and then declines. This suggests that as K increases, more information from neighbors can be captured, leading to an initial improvement. However, as K continues to increase, information from less relevant neighbors is also incorporated, negatively affecting the model's performance. The model achieves its best performance when K is set to 4.

b: ANALYSIS OF EMBEDDING SIZE

In order to assess the impact of the embedding dimension parameter in the model, we investigate the sensitivity of the model to the embedding dimension in Figures. 8. The x-axis represents the embedding dimension, which we set to 8, 16, 32, 64, 128, and 256. The y-axis represents the $F1$ score for multi-class classification tasks.

As shown in Fig. 8, the performance of HGNN-QSSA improves as the embedding dimension increases. The model achieves its best performance when the embedding dimension is around 128. However, when the embedding dimension

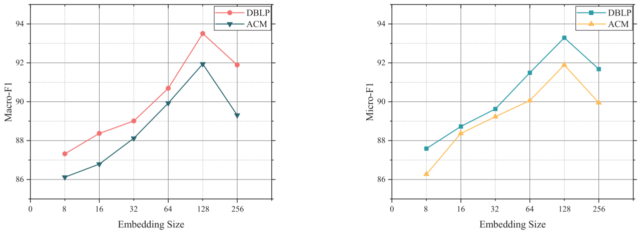


FIGURE 8. Parameters analysis over DBLP and ACM on Embedding dimensions.

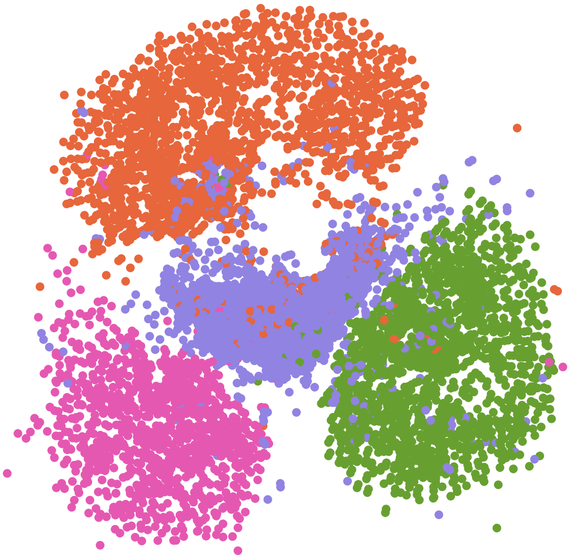


FIGURE 9. Visualization embeddingon DBLP. Each point indicates one paper.

increases to 256, the model's performance decreases. This could occur because a very high embedding dimension may include some irrelevant information, and the aggregation of these features may introduce noise. Therefore, we chose 128 as the embedding dimension.

F. VISUALIZATION

To validate the effectiveness of the node embeddings generated by our proposed HGNN-QSSA model, we conducted visualization of these embeddings. In the DBLP dataset, we utilized t-SNE [44] to reduce the dimensionality of the 128-dimensional node representation to a 2-dimensional space. Visualization is performed for the paper nodes (DBLP-P). Fig. 9 displays the visualization for DBLP-P. We label the nodes in the visualization and used different colors to distinguish between different labels. In the DBLP dataset, authors and papers are categorized into four domains. As shown in Fig. 9, we group papers into four clusters in the 2D space, corresponding to the four domains of papers. Furthermore, each cluster is distinctly separated from the others, with clear boundaries between them, providing further evidence of the effectiveness of the HGNN-QSSA model and the methodology we have proposed.

VI. CONCLUSION

This article introduces an advanced method for heterogeneous graph neural networks. The method is primarily divided into two modules: one is the Quantitative Sampling Module, which selects more representative neighbors to better express node features and reduce noise interference. The second module is the Structure-aware Attention Module. Unlike the traditional approach that only considers attributes, we choose to incorporate structural relevance into the network, enabling a more comprehensive embedding of node information. The experimental data indicates that the algorithm proposed in this paper significantly outperforms other compared algorithms on the DBLP and ACM datasets. However, it's important to acknowledge certain limitations of our model. For instance, its capacity to generalize across dynamic network datasets might be limited due to the potential fluctuation in attributes and relationships among nodes over time. This dynamism inherent in dynamic networks presents a challenge for our model's robustness and adaptability.

Our future work involves tackling large-scale problems, such as scenarios where graphs comprise a significant number of nodes. Furthermore, we will also strive to enhance the model's generalization capability, enabling it to be applied to other heterogeneous information networks. Additionally, we plan to explore novel techniques for optimizing the model's performance on dynamic network datasets, considering the evolving nature of network attributes and relationships. Moreover, we aim to conduct extensive empirical studies to validate the scalability and effectiveness of our approach across diverse real-world applications.

REFERENCES

[1] V. Satuluri, Y. Wu, X. Zheng, Y. Qian, B. Wichers, Q. Dai, G. M. Tang, J. Jiang, and J. Lin, "SimClusters: Community-based representations for heterogeneous recommendations at Twitter," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 3183–3193.

[2] M. Lu, Z. Zhang, Z. Qu, and Y. Kang, "LPANNI: Overlapping community detection using label propagation in large-scale complex networks," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 9, pp. 1736–1749, Sep. 2019.

[3] W. Luo, D. Zhang, L. Ni, and N. Lu, "Multiscale local community detection in social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 1102–1112, Mar. 2021.

[4] S. Yassine, S. Kadry, and M. Sicilia, "Detecting communities using social network analysis in online learning environments: Systematic literature review," *WIREs Data Mining Knowl. Discovery*, vol. 12, no. 1, p. e1431, Jan. 2022.

[5] Z. Zhang, Q. Jiao, Y. Zhang, B. Liu, Y. Wang, and J. Li, "OTUCD: Unsupervised GCN based metagenomics non-overlapping community detection," *Comput. Biol. Chem.*, vol. 98, Jun. 2022, Art. no. 107670.

[6] X. Wang, H. Ji, C. Shi, B. Wang, P. Cui, P. Yu, and Y. Ye, "Heterogeneous graph attention network," in *Proc. World Wide Web Conf.*, 2019, pp. 2022–2032.

[7] S. Zhou, J. Bu, X. Wang, J. Chen, and C. Wang, "HAHE: Hierarchical attentive heterogeneous information network embedding," 2019, *arXiv:1902.01475*.

[8] 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.

[9] Q. Zhao, C. Yu, J. Huang, J. Lian, and D. An, "Sentiment analysis based on heterogeneous multi-relation signed network," *Mathematics*, vol. 12, no. 2, p. 331, Jan. 2024.

[10] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proc. Web Conf.*, New York, NY, USA, Apr. 2020, pp. 2704–2710.

[11] Q. Lv, M. Ding, Q. Liu, Y. Chen, W. Feng, S. He, C. Zhou, J. Jiang, Y. Dong, and J. Tang, "Are we really making much progress? Revisiting, benchmarking and refining heterogeneous graph neural networks," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 1150–1160.

[12] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 793–803.

[13] X. Yang, M. Yan, S. Pan, X. Ye, and D. Fan, "Simple and efficient heterogeneous graph neural network," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 9, pp. 10816–10824.

[14] Q. Zhao, C. Wang, and C. Jiang, "HSim: A novel method on similarity computation by hybrid measure," in *Proc. 6th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2015, pp. 160–165.

[15] D. Nazir, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "SemAttNet: Toward attention-based semantic aware guided depth completion," *IEEE Access*, vol. 10, pp. 120781–120791, 2022.

[16] K. Zhang, M. Liu, J. Zhang, and Z. Dong, "PA-MVSNet: Sparse-to-dense multi-view stereo with pyramid attention," *IEEE Access*, vol. 9, pp. 27908–27915, 2021.

[17] Q. Zhao, Y. He, C. Jiang, P. Wang, M. Qi, and M. Li, "Integration of link and semantic relations for information recommendation," *Comput. Informat.*, vol. 35, no. 1, pp. 30–54, 2016.

[18] Q. Zhao, C. Wang, P. Wang, M. Zhou, and C. Jiang, "A novel method on information recommendation via hybrid similarity," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 3, pp. 448–459, Mar. 2018.

[19] G. Ritzer, *The Blackwell Encyclopedia of Sociology*, vol. 1479. Malden, MA, USA: Wiley-Blackwell 2007.

[20] S. Li, L. Jiang, X. Wu, W. Han, D. Zhao, and Z. Wang, "A weighted network community detection algorithm based on deep learning," *Appl. Math. Comput.*, vol. 401, Jul. 2021, Art. no. 126012.

[21] X. Zhao, J. Liang, and J. Wang, "A community detection algorithm based on graph compression for large-scale social networks," *Inf. Sci.*, vol. 551, pp. 358–372, Apr. 2021.

[22] F. Ye, C. Chen, and Z. Zheng, "Deep autoencoder-like nonnegative matrix factorization for community detection," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 1393–1402.

[23] Y. Xie, M. Gong, S. Wang, and B. Yu, "Community discovery in networks with deep sparse filtering," *Pattern Recognit.*, vol. 81, pp. 50–59, Sep. 2018.

[24] Y. Ma, G. Liu, Y. Ma, and Q. Chen, "Integrative analysis for identifying co-modules of microbe-disease data by matrix tri-factorization with phylogenetic information," *Frontiers Genet.*, vol. 11, p. 83, Feb. 2020.

[25] S. E. Garza and S. E. Schaeffer, "Community detection with the label propagation algorithm: A survey," *Phys. A, Stat. Mech. Appl.*, vol. 534, Nov. 2019, Art. no. 122058.

[26] I. Stanton and G. Kliot, "Streaming graph partitioning for large distributed graphs," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2012, pp. 1222–1230.

[27] D. S. Hochba, "Approximation algorithms for NP-hard problems," *ACM Sigact News*, vol. 28, no. 2, pp. 40–52, 1997.

[28] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell Syst. Tech. J.*, vol. 49, no. 2, pp. 291–307, Feb. 1970.

[29] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113.

[30] A. Ganesh, L. Massoulie, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Societies*, Mar. 2005, pp. 1455–1466.

[31] Y. Liu, X. Wang, S. Wu, and Z. Xiao, "Independence promoted graph disentangled networks," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 4916–4923.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[33] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on convolutional neural networks (CNN) in vegetation remote sensing," *ISPRS J. Photogramm. Remote Sens.*, vol. 173, pp. 24–49, Mar. 2021.

[34] N. Fu, Q. Zhao, Y. Miao, B. Zhang, and D. Wang, "Representation learning method of graph convolutional network based on structure enhancement," *Comput. Informat.*, vol. 41, no. 6, pp. 1563–1588, 2022.

[35] Q. Zhao, G. Liu, F. Yang, R. Yang, Z. Kou, and D. Wang, "Self-supervised signed graph attention network for social recommendation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2023, pp. 1–9.

[36] Q. Zhao, J. Huang, G. Liu, Y. Miao, and P. Wang, "A multiinterest and social interest-field framework for financial security," *IEEE Trans. Computat. Social Syst.*, early access, pp. 1–11, Mar. 2023.

[37] 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.

[38] J. Zhang, Y. Dong, M. Kuang, B. Liu, B. Ouyang, J. Zhu, H. Wang, and Y. Meng, "The art of defense: Letting networks fool the attacker," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 3267–3276, 2023.

[39] D. Jin, Z. Liu, W. Li, D. He, and W. Zhang, "Graph convolutional networks meet Markov random fields: Semi-supervised community detection in attribute networks," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 33, 2019, pp. 152–159.

[40] H. Hong, H. Guo, Y. Lin, X. Yang, Z. Li, and J. Ye, "An attention-based graph neural network for heterogeneous structural learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 4132–4139.

[41] P. Velič ković, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

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

[43] H. Wei, G. Xiong, Q. Wei, W. Cao, and X. Li, "Structure-aware attributed heterogeneous network embedding," *Knowl. Inf. Syst.*, vol. 65, no. 4, pp. 1769–1785, Apr. 2023.

[44] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

QIN ZHAO (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Technology, Tongji University, Shanghai, China, in 2016. He is currently an Associate Professor with the Department of Computer Science and Technology, Shanghai Normal University, Shanghai. He is also the Deputy Director of the Shanghai Engineering Research Center of Intelligent Education and Big Data. He has published more than 30 papers in premier international journals and conferences, including IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS and IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS. His research interests include new-generation artificial intelligence technologies, social network analysis, data mining, and intelligence in scientific computing. He is a Senior Member of the China Computer Federation (CCF).

YARU MIAO received the B.E. degree from the Department of Computer, Shanghai Normal University, Shanghai, China, in 2021, where she is currently pursuing the M.S. degree with the Department of Computer Science and Technology. Her research interests include community detection, natural language processing, and sentiment analysis.

DONGDONG AN received the B.S. and Ph.D. degrees in software engineering from East China Normal University, Shanghai, China, in 2013 and 2020, respectively. She is currently a Lecturer with the Department of Computer Science and Technology, Shanghai Normal University, Shanghai. Her current research interests include model-driven architecture, machine learning, formal methods, and statistical model-checking techniques.

JIE LIAN received the Ph.D. degree from Towson University, in 2017. She is currently an Associate Professor with the Department of Computer Science and Technology, Shanghai Normal University, where she has been a Faculty Member, since 2017. Her research interests include spatio-temporal data mining, deep learning, and big data, ranging from theory to design to implementation. She received the Sailing Talent Program of China, in 2019.

MAOZHEN LI received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 1997. From 1999 to 2002, he was a Postdoctoral Researcher with the School of Computer Science and Informatics, Cardiff University, Cardiff, U.K. He is currently a Professor with the Department of Electronic and Computer Engineering, Brunel University London, London, U.K. His research interests include high-performance computing, including cloud computing and edge computing, big data analytics, and intelligent systems with applications in smart grids and smart cities.