A Heterogeneous Graph Neural Network With Attribute Enhancement and Structure-Aware Attention

Shenghang Fan, Guanjun Liu, *Senior Member, IEEE*, and Jian Li

***Abstract*— Heterogeneous information network (HIN) has been applied in a wide variety of graph analysis tasks. At present, it is a trend of heterogeneous graph neural networks (HGNNs) to cast the meta-paths aside, since it solves the problem of structural information loss caused by artificially designed meta-paths. However, existing meta-path-free HGNNs fail to take into account that most node types in many HINs have no attributes, and they cannot make full use of sparse node attributes when applied to HINs with missing attributes. Furthermore, their computation of attention coefficients explores the correlations of node attributes while almost ignoring structural ones, which may limit the expression ability of the model and cause overfitting in model training. To alleviate these issues, we propose an HGNN with attribute enhancement and structure-aware attention (HGNN-AESA). First, we design an attribute enhancement module (AEM) to connect more useful attributed nodes to the target nodes. Specifically, AEM introduces a random walk with restart (RWR) strategy to obtain structural relevance scores of each node within its specific subgraph. The structural relevance scores are used to capture potentially influential attributed nodes in high-order neighborhood for each target node. Second, we propose heterogeneous structure-aware attention layers (HSALs) to learn node representations. HSALs follow a hierarchical attention framework, including node-level and type-level attention. The node-level attention aggregates feature (attribute) embeddings of same-type neighbors, and the relevant attention coefficients depend on the combination of node attributes and heterogeneous structural interventions. The type-level attention fuses all type-specific vector representations and generates the ultimate node embedding. Finally, extensive experiments on three different real-world HIN datasets demonstrate that our model outperforms state-of-the-art methods.**

***Index Terms*— Graph neural networks (GNNs), heterogeneous information networks (HINs), network representation learning.**

I. Introduction

HETEROGENEOUS information networks (HINs) (a.k.a. heterogeneous graphs), which consist of different types of entities and relationships, have become ubiquitous in real-world scenarios. For example, social media websites contain many types of objects (such as users, posts, and tags) and complex interactions between these objects (such as the publishing relationship between users and posts, the index relationship between posts and tags). Compared with the traditional homogeneous information networks, HINs can describe real-world objects and their interactions more accurately and realistically. Therefore, HINs have been applied in a wide variety of graph analysis tasks, such as link prediction [13], [29], [45], [51], recommendation [27], [31], [32], [36] and node classification [3], [24], [37].

In recent years, graph neural networks (GNNs) have demonstrated ground-breaking performance for processing graph data, such as graph convolutional network (GCN) [22], graph sample and aggregate (GraphSAGE) [13], and graph attention network (GAT) [42]. However, these methods are all designed for homogeneous graphs which only extract part of the information of the actual interactive system, or do not distinguish the differences of objects or relationships [10], [50]. Thus, when applied to heterogeneous graphs, these methods cannot show good performance [46], [50]. To address this limitation, some heterogeneous GNNs (HGNNs) have been developed, such as heterogeneous graph attention network (HAN) [46], hierarchical attentive heterogeneous information network embedding (HAHE) [52],3 and metapath aggregated GNN (MAGNN) [9]. They first use artificially preselected meta-paths to construct homogeneous subgraphs, and then leverage an attention mechanism to learn node representations. In this way, the traditional GNNs can be adapted to HINs and preserve rich semantic information in HINs simultaneously. Despite the success of meta-path-based HGNNs, they cannot manually exploit all the meta-paths that might be useful for downstream tasks, which risks losing important structural information. For this defect, a series of meta-path-free HGNNs have been proposed, such as heterogeneous graph structural attention neural network (HetSANN) [17], heterogeneous graph transformers (HGT) [18], and Simple-HGN [33]. They cast the meta-paths aside and capture heterogeneous structural information of HINs by stacking multiple improved attention layers. Although these meta-path-free methods can preserve structural information more comprehensively than the meta-path-based methods, they still suffer from two issues. First, these methods all follow a neighborhood aggregation framework, which updates the node embeddings by aggregating the first-order neighbors' node features, so whether these node features contain enough useful information greatly affects the performance of the models. However, in many HINs, only a few types of nodes have real attributes [20], which leads to the fact that the input features of most types of nodes are

Manuscript received 20 September 2022; revised 4 December 2022 and 3 January 2023; accepted 14 January 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62172299, in part by the Shanghai Science and Technology Committee under Grant 22511105500, and in part by the Key Research and Development Projects in Zhejiang Province under Grant 2022C02009. *(Corresponding authors: Jian Li; Guanjun Liu.)*

Shenghang Fan and Jian Li are with the School of Mathematics and Computer Science, Zhejiang A&F University, Hangzhou 311300, China (e-mail: fshang1818@163.com; lijian0120@foxmail.com).

Guanjun Liu is with the Key Laboratory of Embedded System and Service Computing of Ministry of Education, Tongji University, Shanghai 201804, China (e-mail: liuguanjun@tongji.edu.cn).

Digital Object Identifier 10.1109/TCSS.2023.3239034

usually randomly generated. For example, in the Association for Computing Machinery (ACM[1]) dataset, the paper nodes have real attributes, but the author and subject nodes are no-attributes because the cost is expensive or even impossible. Therefore, when applied to such HINs, the above methods may not be able to aggregate enough useful information for the target nodes and lead to poor performance. Although stacking more layers can alleviate the problem, this will result in oversmoothing [35], [43] and high computation and memory costs. Second, different from the above HINs with missing attributes, some HINs have complete attributes, such as miRNA-disease network [49]. However, these HGNNs still have a defect when applied to such HINs. Their computation of attention coefficients depends heavily on node attributes without exploring more structural relevance of nodes, which may limit the predictive power of models and lead to overfitting during model training. This is mainly because of an important characteristic of graph structured data, that is, the structures and attributes come from two heterogeneous spaces. Thus, the attention coefficients that consider attributes while almost ignoring structures may cause the models to fail to correctly distinguish some graph structures [14].

One intuitive way to solve the first issue is to enlarge the receptive field of the models, i.e., connect appropriate attributed nodes in high-order neighborhood for nodes. In this way, the models can aggregate more useful attribute information for nodes to improve the performance of node embeddings. At the same time, due to direct connection, the propagation of node features does not require intermediate nodes, which reduces the loss of attribute information during message-passing. However, how to capture the appropriate attributed nodes is an urgent problem that needs to be solved. For second issue, He et al. [14] and Wang et al. [43] incorporate structural intervention into the computation of attention coefficients and improve the expressive power of the models. However, these methods are designed for homogeneous graphs and cannot be directly applied to HINs (heterogeneous graphs). In particular, the heterogeneity of HINs leads to richer structural information and more complex interactions, so how to effectively capture the structures of HINs and incorporate them into the computation of attention coefficients is particularly necessary.

In this article, we propose an HGNN with attribute enhancement and structure-aware attention (HGNN-AESA) to alleviate the above issues. HGNN-AESA includes two components, attribute enhancement module (AEM) and heterogeneous structure-aware attention layers (HSALs). They are elaborated as follows.

1) *AEM:* AEM aims to capture more potentially useful attributed nodes for each target node through graph topology. Generally, we provide node attributes for target node type (i.e., the type of nodes that needs to be analyzed) as much as possible to predict the results more accurately. Thus, in this case (i.e., the target node type have attributes), we only need to search some appropriate high-order homogeneous nodes for each target node to enhance attributes. In a social network, social influence can push people and their current friends toward uniformity of behavior [5]. Based on this, we guess that homogeneous nodes with latent relationships in HINs tend to have similar topological structures through the potential influence between them. For example, in an academic network (Fig. 1), $A1$ and $A2$ copublish multiple papers exhibiting similar structure, which may be influenced by each other through their latent relationships (such as tutor–student or colleague relationship). Motivated by the above inference, we connect some homogeneous nodes with high structural similarity to target nodes to perform attribute enhancement. We refer to these top-k high-order homogeneous nodes as "*potential neighbors*." In addition, when the target nodes' attributes are not available, we perform the above *attribute enhancement* strategy for attributed node types instead of target node type. In this case, our model can still efficiently capture more node attributes for each target node by stacking a few layers.
2) *HSALs:* We present HSALs to improve the expressive power of HGNNs and alleviate overfitting which may occur in the following cases: a) in AEM, we connect too many attributed nodes to target nodes and b) all or most types of nodes in the HINs have attributes. Specifically, HSALs first use type-specific projection matrices to project each neighbor into the same low-dimensional vector space as the target nodes. Subsequently, different from previous graph attentions that rely solely on node attributes, HSALs incorporate heterogeneous structural correlations into the computation of attention coefficients for weighted aggregation of neighbors' attributes and then obtain the node embeddings. At the same time, HSALs can adaptively adjust the relative significance of structural and attribute correlations according to different HINs.

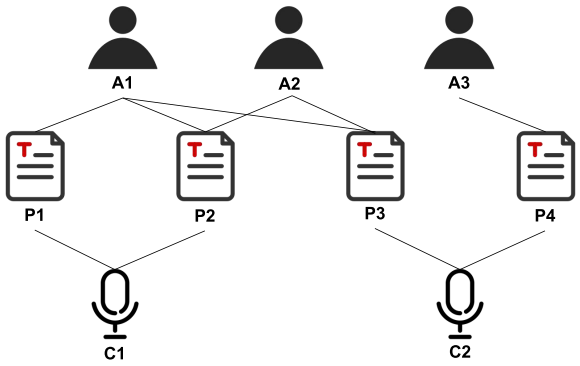


Fig. 1. Illustrative example of an HIN (academic network).

To sum up, the main contributions of our work are as follows.

1) We propose an AEM to enable the target nodes have a larger receptive field in each layer to capture more attributed nodes. Benefiting from this, our model can make full use of the sparse node attributes.
2) We present HSALs, which encode heterogeneous structural information and incorporate it into the computation of content-based (i.e., attribute) attention coefficients. Due to the ability of HSALs to adaptively adjust the

[1]http://dl.acm.org/

relative significance of structural and attribute correlations, our model alleviates overfitting and achieves stronger predictive ability in different HINs.

3) Our model is evaluated with the node classification task on three different real-world HIN datasets. The experimental results show the superiority of our model compared with various state-of-the-art methods.

II. Related Work

In this section, we review the related work in three aspects: 1) HIN embedding; 2) GNNs; and 3) HGNNs.

A. HIN Embedding Methods

The HIN embedding methods mainly focus on preserving the heterogeneous graph structures efficiently and properly. On one hand, some methods [2], [11], [34], [39], [40], [48] aim to preserve different types of first-order proximity between nodes connected by different links. Projected metric embedding (PME) [2] and embedding of embedding (EOE) [48] use relation-specific matrices to project two heterogeneous nodes connected by one link into the same metric space, and then capture the heterogeneity of the graph by minimizing the distance between them or maximizing their similarity; heterogeneous information network embedding via edge representations (HEER) [39] and embedding learning by aspects in heterogeneous information networks (AspEM) [38] capture the heterogeneity by maximizing a heterogeneous similarity function, which constructs the typed closeness between the existing links and their corresponding types. On the other hand, unlike the above methods which only use the first-order information of heterogeneous graphs, Metapath2vec [7], Spacey [15], hyperbolic heterogeneous information network embedding (HHNE) [47], jump and stay strategies (JUST) [19], and balancing the type of relation in heterogeneous information network (BHIN2vec) [25] preserve high-order proximity via random walk variants. Most of them [7], [15], [25], [47] adopt meta-paths, which need to artificially preselection, to guide random walks. In contrast, JUST [19] proposes a random walk method that does not involve any meta-path, which selects the next node through Jump or Stay strategies. However, the HIN embedding methods can only learn the network structures and cannot exploit node attributes.

B. Graph Neural Networks

As an emerging technology that realizes the effective combination of graph data and deep learning technologies, GNNs have received extensive attention from researchers. Inspired by convolutional neural networks, Bruna et al. [1] first apply a convolution operation to graph data and propose a graph convolution network model in the spectral domain. Based on this, many improved spectral-based models [6], [16], [22], [26] have been proposed. Compared with the spectral-based methods, the spatial-based methods [13], [42] have higher computational efficiency, which have been proposed and popularized in recent years. Among them, GAT [42] proposes the masked self-attention mechanism to assign different importance to different nodes within the first-order neighborhood during the convolution operation. Subsequently, to improve the performance of GATs, Duan et al. [8], Klicpera et al. [23], and Wang et al. [44] examine high-order neighborhoods in every layer to relieve the oversmoothing problem caused by the deep GNN architectures, He et al. [14] propose conjoint attention that can incorporate node features and structures to compute more appropriate attention coefficients. However, the above methods are all designed for homogeneous graphs, and thus they are either only applicable to homogeneous graphs [6], [13], [16], [22], [26] or cannot preserve heterogeneous structures when directly applied to heterogeneous graphs [8], [14], [42], [44].

C. Heterogeneous Graph Neural Networks

Recently, since the GNNs directly applied to HINs do not perform well, a number of efforts aim to design HGNNs. HAN [46] and HAHE [52] aggregate the information of high-order homogeneous nodes connected by artificially preselected meta-paths. Based on [46], MAGNN [9] also encodes the intermediate node features along the path instances of a meta-path. Higher-order attribute-enhancing framework (HAE) [28] combines meta-paths and meta-graphs to simultaneously examine first-order and high-order neighborhoods, and then uses self-attention to explore the interactions between a target node and its neighbors. heterogeneous GNN (HetGNN) [50] first leverages a random walk with restart (RWR) to sample a fixed number of neighbors for the target node based on the neighbors' node types, and then learns the importance of different types of neighbors through an attention mechanism to obtain the final node embeddings. HetSANN [17], HGT [18], and Simple-HGN [33] extend GAT [42] to HINs. Specifically, they use self-attention to directly aggregate first-order neighbors, and then capture more global heterogeneous neighborhood information by stacking multiple attention layers. HGNN-AC [20] first uses a meta-path-based embedding method to get topological embeddings which can guide attribute completion for no-attribute nodes, and then combines with arbitrary HGNN models for final node embedding. However, the above methods either are limited by artificially designed meta-paths [9], [28], [28], [46], [52] or cannot make full use of sparse node attributes and use traditional graph attentions that rely solely on node features [17], [18], [33], [50].

III. Preliminaries

In this section, we first introduce the definition of HINs and then formally define the problem of heterogeneous graph representation learning.

A. Definition 1: Heterogeneous Information Network

An HIN is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$ with a node-type mapping function ϕ: $\mathcal{V} \rightarrow \mathcal{A}$ and an edge-type mapping function $\psi : \mathcal{E} \rightarrow \mathcal{R}$, where $\mathcal{V}$ is the sets of nodes, $\mathcal{E}$ is the sets of edges, and $\mathcal{A}$ and $\mathcal{R}$ are the sets of node types and edge types, respectively. Each HIN has different types of nodes or edges such that $|\mathcal{A}| + |\mathcal{R}| > 2$.

B. Definition 2: HIN Representation Learning

Given an HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$and the task $\mathcal{T}$. HIN representation learning aims to exploit the rich structural proximities and semantic information in $\mathcal{G}$ to learn the d-dimensional node representations $h_v \in \mathcal{R}^d$ for all $v \in \mathcal{V}$ with $d \ll |\mathcal{V}|$ that are useful for $\mathcal{T}$.

IV. Model

In this section, we propose an (HGNN-AESA). Fig. 2 shows the overall architecture of our model. In the following, we will walk through the two key components, including AEM and HSALs.

A. Attribute Enhancement Module

We believe that potential neighbors tend to have great influence on the target nodes, so we directly connect potential neighbors for target nodes to perform attribute enhancement. To find these potential neighbors, we use RWR [41] which can explore the local structural details of each node, to quantify the structural proximity between the target node and its homogeneous nodes.

Specifically, we perform a random walk in a subgraph g_i, which consists of node i and its n-order neighborhood. The walk iteratively travels to each neighbor with a probability proportional to edge weight, and at each step it also has a certain probability to return to the starting node i. The iterative formula can be expressed as follows:

$$\vec{r}_i = c\tilde{W}_i\vec{r}_i + (1-c)\vec{e}_i \tag{1}$$

where c is a constant in the range (0,1) which controls the restart probability, $\tilde{W}_i$ is a transition probability matrix by normalizing columns of adjacency matrix W_i where $\tilde{W}_i[i, j]$ represents the probability of walking from j to i, $\vec{e}_i$ is an initial vector whose ith element is 1, and the rest of the elements are 0, $\vec{r}_i$ denotes the structural relevance scores (i.e., steady-state probabilities) of all the neighbors in the subgraph g_i with respect to starting node i, and it can be written in closed form as

$$\vec{r}_i = (1-c)\left(I - c\tilde{W}_i\right)^{-1}\vec{e}_i. \tag{2}$$

By the same method as calculating $\vec{r}_i$, we finally obtain the structural relevance scores of each node in the HIN. To more reasonably use the relevance scores to evaluate the structural proximity between node i and its homogeneous node j, we expand $\vec{r}_i$ to $\vec{r}'_i$ by filling zeros, where $\vec{r}'_i$ denotes the relevance scores of the subgraph $g_i \cup g_j$ of node i. Moreover, there are variety types of neighbors in a subgraph, and if we do not distinguish the types and directly use all the neighbors to calculate structural proximity, it will be difficult for the types of neighbors with a very small number of nodes to affect the results. Therefore, the final calculation result s_{ij} is determined by averaging the structural proximity of each type of neighbors, and it can be obtained as follows:

$$s_{ij}^{\phi_A} = \frac{\Sigma_{v\in\phi_A}\min\left(r'_{iv}, r'_{jv}\right)}{\Sigma_{v\in\phi_A}\max\left(r'_{iv}, r'_{jv}\right)}, s_{ij} = \frac{1}{|\mathcal{A}_{ij}|}\sum_{\phi_A\in\mathcal{A}_{ij}} s_{ij}^{\phi_A} \tag{3}$$

where ϕ_A is the set of A-type neighbors, $\mathcal{A}_{ij}$ is the sets of neighbor node types in $g_i \cup g_j$, and $r'_{iv}$and r'_{jv} represent the vth elements in $\vec{r}'_i$ and $\vec{r}'_j$, respectively. Fig. 3 shows the evaluation of structural proximity between node i and j. Subsequently, by evaluating the structural proximity between node i and different homogeneous nodes, we can pick out top-k homogeneous nodes with the highest structural similarity for node i as potential neighbors. And then we connect these potential neighbors to their target nodes to complete attribute enhancement.

Furthermore, considering that in some HINs with missing attributes, only other types of nodes have attributes rather than the target node type. Thus, we perform the above attribute enhancement strategy for attributed heterogeneous node types. Since these heterogeneous attributed nodes are usually directly connected to the target nodes, our model can still efficiently capture more node attributes for each target node by stacking a few layers.

B. Heterogeneous Structure-Aware Attention Layers

We propose HSALs to preserve the heterogeneous structure of HINs and alleviate the overfitting caused by traditional graph attentions. HSALs incorporate heterogeneous structures into the computation of attention coefficients, which mainly includes two aspects: 1) different from HetSANN and Simple-HGN which directly apply the softmax to normalize all the first-order neighbors to compute the attention coefficients without considering the heterogeneity of neighbors, we use the hierarchical attention structure to first learn the weight of each neighbor among same-type neighbors (node-level attention) and then learn the weight of each type of neighbors (type-level attention) and 2) we incorporate the structural relevance scores which are obtained from AEM into the computation of node-level attention. In each layer, HSALs have three steps, including transformation operation, node-level attention, and type-level attention.

1) Transformation Operation: In HINs, different types of nodes are located in different feature spaces and tend to have unequal dimensions of feature vectors. Therefore, when a target node needs to aggregate heterogeneous neighbors, we cannot calculate directly.

To address this limitation, we design a transformation operation to project the representations of different types of neighbors into a common vector space. For a target node $i \in \mathcal{V}$ of type ϕ_i, we apply a type-specific transformation matrix $W_{\phi_i,\phi_j}^{l+1} \in \mathbb{R}^{n_{\phi_i}^{l+1}\times n_{\phi_j}^{l}}$ to project each j-type neighbor of node i to the same latent factor space, as

$$h_{\phi_i,j}^{l+1} = W_{\phi_i,\phi_j}^{l+1} h_j^l \tag{4}$$

where $h_{\phi_i,j}^{l+1}$ is the feature vector of node j of layer $l+1$ in i-type space, obtained by projecting the feature vector h_j^l of previous layer. And we take the input feature vector as the cold start state h_i^0 for node i.

2) Node-Level Attention: Node-level attention can learn the weight of neighbors belonging to same type. Here, we treat the potential nodes as nodes of a different type from homogeneous nodes in the first-order neighborhood. This is because the first-order homogeneous nodes have a real association with the target node, but the relationship between potential neighbors and the target node is inferred from the graph structure.

We first use the content-based type-aware attention [17] to evaluate the attention scores between target node i and each node of one type in the receptive field $\mathcal{N}_i$ (i.e., first-order and potential neighbors of node i). For a neighbor $j \in \mathcal{N}_i$ of type

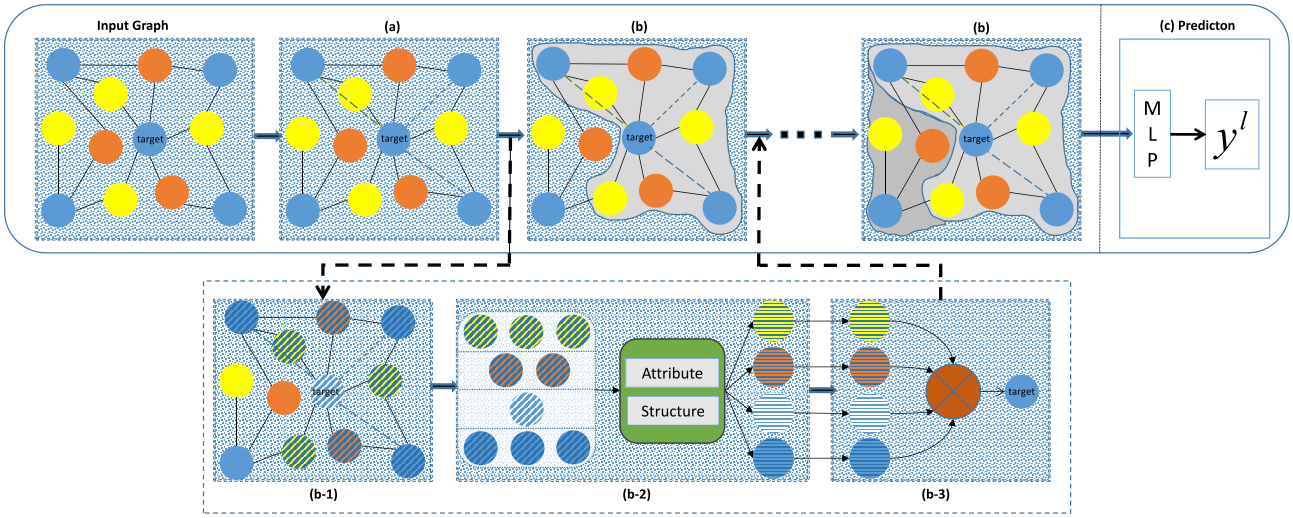


Fig. 2. Overall architecture of the HGNN-AESA model. (a) Target node captures more attribute information through potential neighbors, which are connected by dotted lines. (b) In this figure, we only describe the neighborhood aggregation of the target node through n HSALs. In fact, like GAT, every node in the graph will perform this step. (b-1) All the neighbors in the receptive field are projected into the same vector space. (b-2) Each neighbor is aggregated according to its node type, and the compute of attention coefficients combines structural and attribute correlations. Note that the potential neighbors are not the same type as homogeneous nodes in first-order neighborhood (i.e., different from the node type of the target node, so the potential neighbors and the target node are given different colors in this figure). (b-3) Jointly learn the importance of each type of neighbors and fuse all type-specific vectors to obtain the node embedding of the target node. (c) Calculate the loss and optimization.

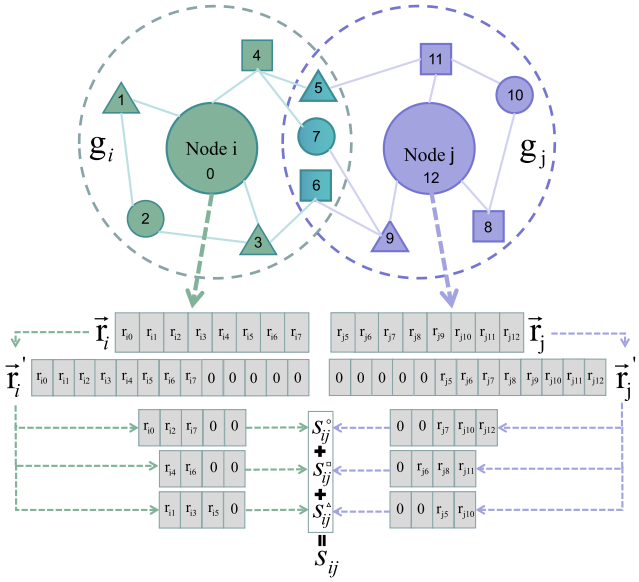


Fig. 3. Explanation of AEM.

ϕ_j, the attention score e_{ij} can be expressed as

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

where $a_r \in \mathbb{R}^{2n_{\phi_i}^{l+1}}$ is a trainable parameter shared by the same edge relation r, and $||$ denotes the concatenation operation. After obtaining the attention score of each neighbor of j-type to target node i, we apply the softmax function to normalize them to get the attention coefficient

$$\alpha_{ij}^{l+1} = \frac{\exp\left(e_{ij}^{l+1}\right)}{\sum_{k\in\phi_j}\exp\left(e_{ik}^{l+1}\right)}. \tag{6}$$

However, the above method for computing attention coefficients mainly uses the content information of neighbors and the structural information is almost ignored. Given that HetGNN [50] exploits the structural information by sampling the neighbors with the highest probability via RWR and achieves good performance, we combine the neighbors' steady-state probabilities (i.e., relevance scores obtained in AEM) with the attention coefficients to compute the importance coefficient. Our algorithm steps are as follows.

Step 1: Normalize the structural relevance scores to obtain structure coefficient β_{ij} as

$$\beta_{ij} = \frac{\exp(r_{ij})}{\sum_{k\in\mathcal{N}_i^{\phi_j}}\exp(r_{ik})}. \tag{7}$$

Note that the above formula only works when j is a first-order neighbor; when j is a potential neighbor, we replace r_{ij} with s_{ij} obtained from AEM. Because we believe that nodes with more similar structures tend to have more great influence on the target node

$$\beta_{ij} = \frac{\exp(s_{ij})}{\sum_{k\in\mathcal{N}_i^{\phi_j}}\exp(s_{ik})}. \tag{8}$$

Step 2: We leverage two learnable parameters p_c^{l+1} and p_s^{l+1} to adaptively adjust the relative significance of node content and structural correlations, denoted as q_c^{l+1} and q_s^{l+1}, respectively. And they are expressed as follows:

$$\begin{aligned} q_c^{l+1} &= \frac{\exp\left(p_c^{l+1}\right)}{\exp\left(p_s^{l+1}\right)+\exp\left(p_c^{l+1}\right)} \\ q_s^{l+1} &= \frac{\exp\left(p_s^{l+1}\right)}{\exp\left(p_s^{l+1}\right)+\exp\left(p_c^{l+1}\right)}. \end{aligned} \tag{9}$$

Then, we combine the attention coefficient α_{ij}^{l+1} and the structure coefficient β_{ij} to compute the node-level

importance coefficient γ_{ij}^{l+1}

$$\gamma_{ij}^{l+1} = \frac{q_s^{l+1}\beta_{ij} + q_c^{l+1}\alpha_{ij}^{l+1}}{\sum_{k\in\mathcal{N}_i^{\phi_j}}\left(q_s^{l+1}\beta_{ik} + q_c^{l+1}\alpha_{ik}^{l+1}\right)}. \tag{10}$$

Step 3: Aggregate all the neighbors of j-type in the receptive field $\mathcal{N}_i$ to obtain the final vector representation of j-type neighbors for node i as follows:

$$H_{i,\phi_j}^{l+1} = \sigma\left(\sum_{k\in\mathcal{N}_i^{\phi_j}} \gamma_{ik}^{l+1} h_{\phi_i,k}^{l+1}\right). \tag{11}$$

To sum up, given a target node $i \in \mathcal{V}$, the node-level attention generates multiple type-specific vector representations of node i, denoted as $\{H_{i,\phi_1}^{l+1}, H_{i,\phi_2}^{l+1}, \ldots, H_{i,\phi_i}^{l+1}, H_{i,\phi_p}^{l+1}\}$, where ϕ_p is the type of potential neighbors, and $\phi_1, \phi_2, \ldots, \phi_i$ are each type of first-order neighbors, respectively. Note that ϕ_i includes node i, so if there are no homogeneous nodes in the first-order neighborhood, H_{i,ϕ_i}^{l+1} depends entirely on node i itself.

3) Type-Level Attention: Generally, nodes in an HIN have multiple types of neighbors, and each type of neighbors can only reflect a one-sided interaction with a target node. To learn more accurate node embeddings, we need to learn the importance of different types of neighbors and fuse their interactions with the target nodes. For j-type neighbors of node i, its type-level attention coefficient $w_i^{\phi_j}$ can be expressed as

$$w_i^{\phi_j} = \frac{\exp\left(\text{Sigmoid}\left(a\left[\text{WH}_{i,\phi_j}^{l+1}, \text{WH}_{i,\phi_i}^{l+1}\right]\right)\right)}{\sum_{p=1}^{p}\exp\left(\text{Sigmoid}\left(a\left[\text{WH}_{i,\phi_p}^{l+1}, \text{WH}_{i,\phi_i}^{l+1}\right]\right)\right)} \tag{12}$$

where W is a learnable weight matrix to perform a linear transformation on each type-specific vector of node i obtained from node-level attention, and a is a single-layer feedforward neural network. And then we fuse all type-specific vector representations to get the final node embedding of node i in this layer

$$h_i^{l+1} = \sum_{p=1}^{p} w_i^{\phi_p} \cdot H_{i,\phi_p}^{l+1}. \tag{13}$$

To perform HIN representation learning, we apply the final node representations obtained from the last layer to different downstream tasks and design task-specific loss functions. For the semisupervised node classification task experimented in this article, we minimize the cross-entropy loss between the ground truth and the predictions

$$\mathcal{L} = -\sum_{l\in\mathcal{V}_L^p} y^l \ln\left(C \cdot h^l\right) \tag{14}$$

where $\mathcal{V}_L^p$ is the set of labeled nodes of the node type ϕ_p, C denotes the parameter of node classifier, and y^l and h^l are labeled node l' ground truth and embedding, respectively.

TABLE I
STATISTICS OF THE DATASETS

Dataset	Node	Number	Features
DBLP	# author (A)	4057	128
	# paper (P)	14328	128
	# conf (C)	20	128
	# term (T)	8789	128
ACM	# paper (P)	4025	128*
	# author (A)	7167	128
	# subject (S)	60	128
IMDB	# movie (M)	3228	14*
	# actor (A)	42553	128
	# director (D)	2103	128
	# user (U)	2016	128

V. EXPERIMENTS

In this section, we use three widely used real-world HINs [database systems and logic programming (DBLP), ACM, and internet movie database (IMDB)] to construct the datasets and conduct extensive experiments to evaluate the performance of our model in a node classification task.

A. Datasets

The simple statistics of the three HIN datasets are summarized in Table I. Note that only the nodes with notation * have available real attributes, and we randomly generate a 128-D vector from the Xavier uniform distribution [12] for each other node to represent its input features.

1) *DBLP:* DBLP is a subset extracted by HAN [46] from a computer science bibliography website.[2] Based on DBLP, we construct an HIN containing four node types: 14 328 papers (P), 4057 authors (A), 20 conferences (C), 8789 terms (T), and the interactions: P $\leftrightarrow$ A, P $\leftrightarrow$ C, and P $\leftrightarrow$ T. The authors are labeled with the four research areas (database, data mining, artificial intelligence, and information retrieval) according to the conferences where their papers are published.
2) *ACM:* The dataset is extracted by Wang et al. [46] from ACM[3] digital library. Based on ACM, we construct an HIN containing three node types: 4025 papers (P), 7167 authors (A), and 60 subjects (S), and the interactions: P $\leftrightarrow$ A and P $\leftrightarrow$ S. Papers are labeled in three areas (database, wireless communication, and data mining) according to the conference they published, and their features correspond the TF-IDF representations of their titles.
3) *IMDB:* We extract a subset from the IMDB dataset in HetRec 2011[4] and construct an HIN containing four node types: 3228 movies (M), 42 553 actors (A), 2016 directors (D), 2103 user (U), and the interactions: M $\leftrightarrow$ A, M $\leftrightarrow$ D, and M $\leftrightarrow$ C. Movies are labeled by four classes (comedy, documentary, drama, and horror) according to their genre, and their features consist of 14 numerical and categorical original features.

[2]https://dblp.org/
[3]https://dl.acm.org/
[4]https://grouplens.org/datasets/hetrec-2011/

B. Baselines

We compare our model against several state-of-art baselines, including one traditional HIN embedding models (Metapath2vec), four HGNN models (HAN, HAHE, DHNE, and HetSANN), and three homogeneous GNN models (GraphSAGE, GAT, and GCN).

1) *Metapath2vec (Mp2vec) [7]:* It is a traditional HIN embedding method. It first designs a meta-path-guided random walk, which mines the semantic information between the target node and its context nodes. It applies the skip-gram to learn node representations. It only leverages node structure information and ignores the node content features.
2) *HAN [46]:* It is the state-of-the-art HGNN model. It constructs several homogeneous subgraphs via given multiple symmetric meta-paths and performs semantic-level attention and node-level attention to learn the importance of nodes and meta-paths.
3) *HAHE [52]:* It is similar to HAN and able to capture the preferences of meta-paths and path instances. Different from HAN, it takes the meta-path-based structural features as the initial feature vector of the nodes.
4) *DHNE [4]:* It is one of the two components (DHNE and AQHN) in ActiveHNE [4], and we did not apply another component to the experiment for a fair comparison. It also considers multiple task-related meta-paths but ignores the importance of different meta-paths.
5) *HetSANN [17]:* It is a heterogeneous method which directly encodes structural information without meta-paths. It aggregates the representations of heterogeneous neighbors by the attention mechanism. Based on the performance as reported, we apply the variant HetSANN.M.R.V to conduct experiment.
6) *GraphSAGE (GSAGE) [13]:* It is a homogeneous GNN method that proposes three different aggregator functions to aggregate node feature information of neighbors. Here, we apply the mean aggregator to conduct experiment.
7) *GAT [42]:* It is a homogeneous GNN method that leverages the masked self-attention mechanism to aggregate the neighborhood information of nodes during the convolution operation.
8) *GCN [22]:* It is a homogeneous GNN method. Unlike GAT and GraphSAGE, it performs convolution operations in the spectral domain rather than the spatial domain.
9) *HGNN-AESA:* It is our proposed semisupervised HGNN which uses one AEM and three heterogeneous structure-aware attention layers.

For implementation details, we use Adam optimizer [21] to optimize our proposed model with a learning rate of 0.005 and randomly initialize parameters with Xavier uniform distribution. For Mp2vec, we set the window size and negative sample size to 5 and the walk length to 100. For meta-path-based methods, including Mp2vec, HAN, and HAHE, we preselect the meta-paths used in the paper [46] for these methods. For methods such as GCN, GAT, and GSAGE that cannot be directly applied to HINs, we first use the above-mentioned meta-paths to construct homogeneous graphs and test them on the homogeneous graphs; we set the number of layers to 3. For models that use a multihead attention mechanism, including HAN, HAHE, HetSANN, and GAT, we set the number of attention heads to 8. For GNNs, including HAN, HAHE, DHNE, HetSANN, GSAGE, GCN, GAT, and HGNN-AESA, we set the dropout rate to 0.6 and train the GNNs for 150 epochs.

C. Node Classification

Node classification can predict the classes of unlabeled nodes based on the existing labeled nodes, which is one of the main tasks to evaluate the performance of HIN models. Here, we leverage the averaged Micro-$F1$ and Macro-$F1$ as evaluation metrics for node classification to compare the performance of all the methods mentioned in the baselines. These two metrics can be formulated as

$$\begin{aligned}
\text{Precision}_{\text{micro}} &= \frac{\sum_{i=1}^{n} \text{TP}_i}{\sum_{i=1}^{n} \text{TP}_i + \sum_{i=1}^{n} \text{FP}_i} \\
\text{Recall}_{\text{micro}} &= \frac{\sum_{i=1}^{n} \text{TP}_i}{\sum_{i=1}^{n} \text{TP}_i + \sum_{i=1}^{n} \text{FN}_i} \\
F1_{\text{micro}} &= 2 \cdot \frac{\text{Precision}_{\text{micro}} \cdot \text{Recall}_{\text{micro}}}{\text{Precision}_{\text{micro}} + \text{Recall}_{\text{micro}}}
\end{aligned} \tag{15}$$

$$\begin{aligned}
\text{Precision}_{\text{macro}} &= \frac{\sum_{i=1}^{n} \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}}{n} \\
\text{Recall}_{\text{macro}} &= \frac{\sum_{i=1}^{n} \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}}{n} \\
F1_{\text{macro}} &= 2 \cdot \frac{\text{Precision}_{\text{macro}} \cdot \text{Recall}_{\text{macro}}}{\text{Precision}_{\text{macro}} + \text{Recall}_{\text{macro}}}.
\end{aligned} \tag{16}$$

Specifically, we perform author classification on DBLP, paper classification on ACM, and movie classification on IMDB. We randomly select $\{20\%, 40\%, 60\%, 80\%\}$ of the nodes as training set, and the rest are equally divided into the validation set and test set. For a fair comparison, all the models use the exact same training, validation, and test sets. To ensure the validity of the test, we repeat the process on each dataset for ten times for each model, Table II illustrates the results.

As shown in Table II, HGNN-AESA is superior to other baselines. The main reason is that HGNN-AESA obtains more accurate node representation through an AEM and structure-aware attention layers that can adapt to various HINs. It is worth noting that HGNN-AESA does not have much superiority than meta-path-based HGNNs on DBLP, even HAHE outperforms our model when the training set is 40%, which may be caused by two reasons. One is that DBLP is a nonattributed HIN, so our model cannot demonstrate its ability to enhance attributes. The other is that on DBLP, we can provide meta-paths directly related to the task (i.e., author classification) for the models. Specifically, the authors are labeled according to the conferences they submitted, which enables HAHE and HAN to achieve very good author classification performance using the meta-path APCPA to establish a neighborhood of authors whose papers have been published in the same conference. Nevertheless, HGNN-AESA still has

TABLE II
QUANTITATIVE RESULTS (%) ON THE NODE CLASSIFICATION TASK

Datasets	Metrics	Training	GSAGE	GCN	GAT	Mp2vec	HAHE	HAN	DHNE	HetSANN	HGNN-AESA
DBLP	Micro F1	20%	88.82	91.55	90.97	90.15	93.57	92.24	84.45	93.36	**93.85**
		40%	88.81	91.10	91.22	90.81	93.61	92.40	84.61	93.72	**94.09**
		60%	88.68	90.48	90.81	89.92	93.65	92.80	86.77	93.85	**94.36**
		80%	88.87	91.72	91.73	90.89	94.38	93.08	87.36	94.03	**95.28**
	Macro F1	20%	87.87	90.6	91.96	90.43	93.11	93.11	83.99	92.82	**93.42**
		40%	87.98	90.17	92.16	89.73	**93.78**	93.30	84.8	92.51	93.75
		60%	88.05	89.73	91.84	90.48	93.45	93.70	86.24	93.17	**93.95**
		80%	88.29	90.99	92.55	90.97	94.24	93.99	86.82	93.48	**94.79**
ACM	Micro F1	20%	81.47	78.80	74.18	66.74	77.17	73.58	76.21	78.57	**81.63**
		40%	80.86	78.64	72.01	69.01	78.19	77.44	78.41	78.62	**82.15**
		60%	80.31	77.78	76.18	71.68	78.09	76.47	78.97	79.25	**81.40**
		80%	81.12	79.75	77.20	73.27	80.86	76.14	79.02	81.13	**82.56**
	Macro F1	20%	63.40	60.19	58.18	50.92	53.87	64.69	64.94	57.89	**65.37**
		40%	62.35	59.12	61.14	51.91	54.82	64.39	65.67	58.80	**66.15**
		60%	60.38	58.80	53.79	51.87	54.65	65.31	65.99	59.20	**67.13**
		80%	59.60	60.25	59.36	54.81	58.84	66.26	67.61	61.08	**68.76**
IMDB	Micro F1	20%	58.20	59.58	55.31	49.87	54.89	56.47	61.69	61.07	**66.02**
		40%	57.11	58.49	55.42	50.14	55.00	56.03	63.03	61.34	**67.68**
		60%	59.89	59.81	55.14	50.83	55.01	57.00	64.16	62.21	**69.38**
		80%	58.27	58.73	54.06	50.90	54.64	57.18	64.85	63.25	**69.75**
	Macro F1	20%	40.93	29.26	30.07	20.95	21.47	45.87	51.27	50.41	**56.15**
		40%	42.68	29.15	31.26	21.36	23.52	44.44	56.22	54.52	**60.10**
		60%	39.64	30.95	29.82	20.89	21.72	44.82	55.95	54.37	**61.20**
		80%	39.22	30.36	28.93	22.11	25.75	44.77	54.92	53.06	**60.60**

a better performance than HetSANN. This is because although the AEM cannot obtain node attributes in such HINs, it enables our model have a larger receptive field than latter and larger receptive field has been proven to be a key performance factor in state-of-the-art neural networks [30].

On ACM and IMDB, as expected, HGNN-AESA achieves more performance gains than on DBLP. This is mainly because the target nodes in both ACM and IMDB have attributes, and our model can make full use of these node attributes and use a more reasonable structure-aware graph attention for attribute aggregation. Especially on IMDB where node attributes are more sparse than ACM, our model outperforms the best baseline (DHNE) by 3%–6%, which indicates that HGNN-AESA has stronger performance than other models in HINs with more sparse node attributes. Furthermore, we can find that HAHE and HAN perform even worse than homogeneous methods on IMDB. Because movies are labeled according to their genre and there are no meta-paths to provide information directly related to the task. This reflects the flaws of the meta-path-based methods.

D. Ablation Study

In this section, we use an ablation experiment to demonstrate that each component of our method has a positive effect on model performance improvement. Specifically, we design several model variants which include: 1) **HGNN-SA** that applies the HSALs without AEM; 2) **HGNN-AE** that applies AEM and uses content-based graph attentions for node aggregation; and 3) **HGNN-AESA**$_{\text{lr}}$ and **HGNN-AESA**$_{\text{th}}$ that apply the HSALs and AEM, but in (12) use activation function LeakyRelu and Tanh, respectively. The results of node classification on IMDB are reported in Fig. 4. From Fig. 4, the following results are obtained.

1) HGNN-AESA outperforms HGNN-SA by 3%–4%, which proves that AEM is effective in using sparse node attributes.
2) HGNN-AESA has a better performance than HGNN-AE, indicating that our proposed structure-aware attention is better than popular graph attention for neighborhood information interactions.
3) HGNN-AE outperforms HGNN-SA. It indicates that providing more attribute information for target nodes may improve the model performance more than improving the aggregation method.
4) Comparing HGNN-AESA with HGNN-AESA$_{\text{th}}$ and HGNN-AESA$_{\text{lr}}$, we see that HGNN-AESA shows the best performance, which reveals that Sigmoid function is more suitable for type-level attention than LeakyRelu and Tanh function.

E. Hyperparameters' Sensitivity

Finally, to explore the effect of the hyperparameters on our proposed model, we conduct several parameters' sensitivity experiments on the IMDB dataset and report the results in Fig. 5.

1) *Number of Potential Neighbors k:* To enable the target nodes to aggregate a suitable number of potential neighbors, we explore the performance of the model when the value of k varies from 10 to 40. The result is shown in Fig. 5(a). According to the result, we can find that the best value of k is around 30, beyond which all the evaluation metrics decrease slowly. This may be because

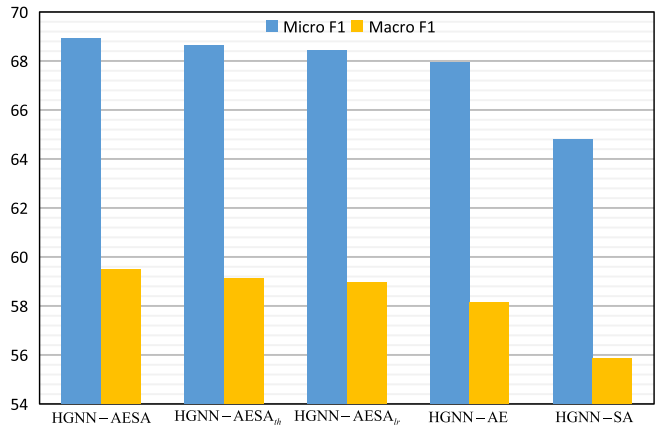


Fig. 4. Performances of variant proposed models on IMDB.

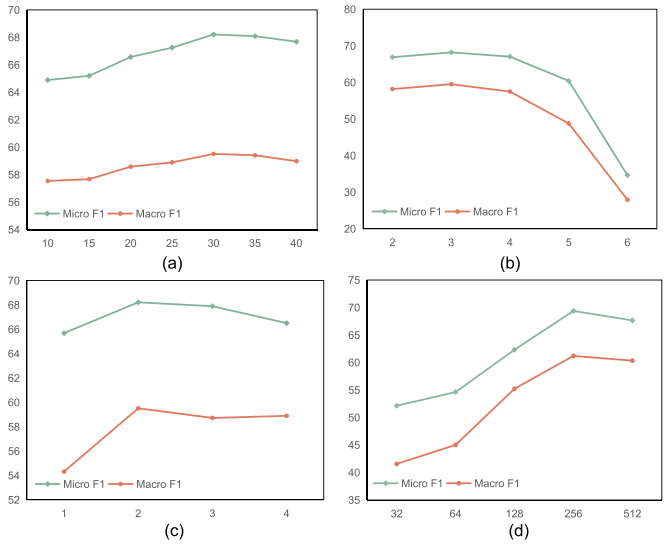


Fig. 5. Hyperparameter analysis of our model on IMDB. (a) Number of potential neighbors. (b) Number of layers. (c) Neighborhood orders. (d) Dimension of the final embedding.

when the value of k is too large, more noise nodes will be involved.

2) *Number of Layers l:* Since the more layers, the more complex neighborhood information obtained by nodes, we explore the different performance of our model when l varies from 2 to 6. From the result which is shown in Fig. 5(b), we can find that the performance of our model raises first and achieves the best performance when $l = 3$, and then starts to go down which may due to oversmoothing issues.
3) *Neighborhood Orders n:* To mine the effective structural information for each node, we explore the performance of the model when the neighborhood order n varies from 1 to 4. The result is shown in Fig. 5(c). As can be seen, when we set n to 2, our model achieves the best performance.
4) *Dimension of the Final Embedding h:* We explore the impact of final node embedding dimensions on model performance. The result is shown in Fig. 5(d). From the results, we can find that the performance of our model first raises with the growth of the embedding dimensions and then slowly drops, which may because too large embedding dimensions result in more additional redundancies.

VI. Conclusion

In this article, we propose a novel HGNN-AESA to make full use of sparse node attributes in HINs with missing attribute. Meanwhile, we use an improved-attention which simultaneously encode heterogeneous structure and attribute correlations to perform more reasonable node aggregation. The experimental results of node classification task on three real-world datasets show that our model outperforms state-of-the-art methods.

References

[1] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*.
[2] H. Chen, H. Yin, W. Wang, H. Wang, Q. V. H. Nguyen, and X. Li, "PME: Projected metric embedding on heterogeneous networks for link prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1177–1186.
[3] T. Chen and Y. Sun, "Task-guided and path-augmented heterogeneous network embedding for author identification," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Feb. 2017, pp. 295–304.
[4] X. Chen, G. Yu, J. Wang, C. Domeniconi, Z. Li, and X. Zhang, "ActiveHNE: Active heterogeneous network embedding," 2019, *arXiv:1905.05659*.
[5] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri, "Feedback effects between similarity and social influence in online communities," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 160–168.
[6] 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. 1–9.
[7] 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*, 2017, pp. 135–144.
[8] R. Duan, C. Yan, J. Wang, and C. Jiang, "Path-aware multi-hop graph towards improving graph learning," *Neurocomputing*, vol. 494, pp. 13–22, Jul. 2022.
[9] X. Fu, J. Zhang, Z. Meng, and I. King, "MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proc. Web Conf.*, Apr. 2020, pp. 2331–2341.
[10] H. Gao, J. Huang, Y. Tao, W. Hussain, and Y. Huang, "The joint method of triple attention and novel loss function for entity relation extraction in small data-driven computational social systems," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 6, pp. 1725–1735, Dec. 2022.
[11] H. Gao, B. Qiu, R. J. Duran Barroso, W. Hussain, Y. Xu, and X. Wang, "TSMAE: A novel anomaly detection approach for Internet of Things time series data using memory-augmented autoencoder," *IEEE Trans. Netw. Sci. Eng.*, early access, Mar. 29, 2022, doi: 10.1109/TNSE.2022.3163144.
[12] 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.
[13] 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.
[14] T. He, Y. S. Ong, and L. Bai, "Learning conjoint attentions for graph neural nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 2641–2653, 2021.
[15] Y. He, Y. Song, J. Li, C. Ji, J. Peng, and H. Peng, "HeteSpaceyWalk: A heterogeneous spacey random walk for heterogeneous information network embedding," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 639–648.
[16] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," 2015, *arXiv:1506.05163*.

[17] 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, 2020, pp. 4132–4139.

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

[19] R. Hussein, D. Yang, and P. Cudré-Mauroux, "Are meta-paths necessary? Revisiting heterogeneous graph embeddings," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 437–446.

[20] D. Jin, C. Huo, C. Liang, and L. Yang, "Heterogeneous graph neural network via attribute completion," in *Proc. Web Conf.*, Apr. 2021, pp. 391–400.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

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

[23] J. Gasteiger, S. Weißenberger, and S. Günnemann, "Diffusion improves graph learning," 2019, *arXiv:1911.05485*.

[24] X. Kong, P. S. Yu, Y. Ding, and D. J. Wild, "Meta path-based collective classification in heterogeneous information networks," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2012, pp. 1567–1571.

[25] S. Lee, C. Park, and H. Yu, "Bhin2Vec: Balancing the type of relation in heterogeneous information network," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 619–628.

[26] R. Levie, M. Federico, X. Bresson, and B. Xavier, "CayleyNets: Graph convolutional neural networks with complex rational spectral filters," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 97–109, Jan. 2019.

[27] J. Li, G. Liu, C. Yan, and C. Jiang, "LORI: A learning-to-rank-based integration method of location recommendation," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 3, pp. 430–440, Jun. 2019.

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

[29] G. Liu, J. Tang, Y. Tian, and J. Wang, "Graph neural network for credit card fraud detection," in *Proc. Int. Conf. Cyber-Phys. Social Intell. (ICCSI)*, Dec. 2021, pp. 1–6.

[30] Z. Liu, W. Liu, P.-Y. Chen, C. Zhuang, and C. Song, "hpGAT: High-order proximity informed graph attention network," *IEEE Access*, vol. 7, pp. 123002–123012, 2019.

[31] W. Luan, G. Liu, C. Jiang, and L. Qi, "Partition-based collaborative tensor factorization for POI recommendation," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 3, pp. 437–446, Jul. 2017.

[32] W. Luan, G. Liu, C. Jiang, and M. Zhou, "MPTR: A maximal-marginal-relevance-based personalized trip recommendation method," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 11, pp. 3461–3474, Nov. 2018.

[33] Q. Lv et al., "Are we really making much progress? Revisiting, benchmarking and refining heterogeneous graph neural networks," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery & Data Mining*, 2021, pp. 1150–1160.

[34] R. Matsuno and T. Murata, "MELL: Effective embedding method for multiplex networks," in *Proc. Companion Web Conf.*, 2018, pp. 1261–1268.

[35] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," 2019, *arXiv:1905.10947*.

[36] A. Salamat, X. Luo, and A. Jafari, "HeteroGraphRec: A heterogeneous graph-based neural networks for social recommendations," *Knowl.-Based Syst.*, vol. 217, Apr. 2021, Art. no. 106817.

[37] L. D. Santos, B. Piwowarski, and P. Gallinari, "Multilabel classification on heterogeneous graphs with Gaussian embeddings," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Riva del Garda, Italy: Springer, 2016, pp. 606–622.

[38] Y. Shi, H. Gui, Q. Zhu, L. Kaplan, and J. Han, "AspEm: Embedding learning by aspects in heterogeneous information networks," in *Proc. SIAM Int. Conf. Data Mining*, 2018, pp. 144–152.

[39] Y. Shi, Q. Zhu, F. Guo, C. Zhang, and J. Han, "Easing embedding learning by comprehensive transcription of heterogeneous information networks," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2190–2199.

[40] J. Tang, M. Qu, and Q. Mei, "PTE: Predictive text embedding through large-scale heterogeneous text networks," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1165–1174.

[41] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *Proc. 6th Int. Conf. Data Mining (ICDM)*, Dec. 2006, pp. 613–622.

[42] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *Stat*, vol. 1050, p. 20, Oct. 2017.

[43] G. Wang, R. Ying, J. Huang, and J. Leskovec, "Improving graph attention networks with large margin-based constraints," 2019, *arXiv:1910.11945*.

[44] G. Wang, R. Ying, J. Huang, and J. Leskovec, "Multi-hop attention graph neural network," 2020, *arXiv:2009.14332*.

[45] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and Q. Liu, "SHINE: Signed heterogeneous information network embedding for sentiment link prediction," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, 2018, pp. 592–600.

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

[47] X. Wang, Y. Zhang, and C. Shi, "Hyperbolic heterogeneous information network embedding," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5337–5344.

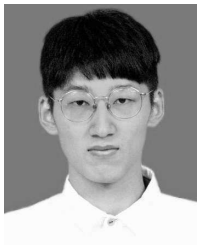
[48] L. Xu, X. Wei, J. Cao, and P. S. Yu, "Embedding of embedding (EOE) joint embedding for coupled heterogeneous networks," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, 2017, pp. 741–749.

[49] P. Xuan, T. Shen, X. Wang, T. Zhang, and W. Zhang, "Inferring disease-associated microRNAs in heterogeneous networks with node attributes," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 17, no. 3, pp. 1019–1031, May 2020.

[50] 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.

[51] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "MetaGraph2Vec: Complex semantic path augmented heterogeneous network embedding," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2018, pp. 196–208.

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

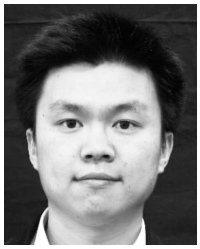


Shenghang Fan is currently pursuing the master's degree with the School of Mathematics and Computer Science, Zhejiang A&F University, Hangzhou, China.

His research interests include machine learning, representation learning, and graph neural network.

Guanjun Liu (Senior Member, IEEE) received the Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2011.

He was a Post-Doctoral Research Fellow with the Singapore University of Technology and Design, Singapore, from 2011 to 2013, and the Humboldt University of Berlin, Berlin, Germany, from 2013 to 2014, supported by the Alexander von Humboldt Foundation. He is currently a Professor with the Department of Computer Science, Tongji University. He has authored over 130 articles and three books. His research interests include Petri net theory, model checking, machine learning, cyber-physical systems, workflow, and credit card fraud detection.



Jian Li was born in Ruian, Zhejiang, China, in 1981. He received the Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2020.

He is currently a Vice Professor with the School of Mathematics and Computer Science, Zhejiang A&F University, Hangzhou, China. His research interests include matrix decomposition, graph neural networks, and their applications.