Computing and Informatics, Vol. 42, 2023, 1281–1304, doi: 10.31577/cai_2023_6_1281

HGCAN: HETEROGENEOUS GRAPH COMPLETION METHOD BASED ON ATTRIBUTE NEIGHBORHOOD

Zhaohui Zhang

School of Computer Science and Technology, Donghua University
Shanghai, 201620, China
&
Shanghai Engineering Research Center of Network Information Services
Shanghai, 201804, China
e-mail: zhzhang@dhu.edu.cn

Siting Huang, Chaochao Hu, Pengwei Wang

School of Computer Science and Technology, Donghua University
Shanghai, 201620, China

e-mail: shelaghst@163.com

Abstract. Recent research shows that the effect of heterogeneous graph embedding learning is vulnerable to non-attribute nodes. However, the existing methods mainly use first-order neighbor nodes to complete attributes, which cannot achieve a satisfactory completion effect on the heterogeneous graphs with random non-attribute nodes. Therefore, this paper put forward an attribute completion method of heterogeneous graphs based on attribute neighborhoods, which is called HGCAN. HGCAN employs two major stages of completion. Specifically, in the first stage, we use metapaths to construct attribute neighborhoods of non-attribute nodes. The attribute neighborhoods aggregation can capture the semantic relations of attributed nodes to initially complete attributes. Then, the second stage uses structural information to obtain the distance relationships between nodes to further improve the preliminary completed attributes. Finally, HGCAN is combined with an existing heterogeneous graph embedding learning model to verify the validity of the completed attributes and make the system end-to-end. Extensive experiments carried out on the ACM dataset show the proposed mechanism's superior performance over state-of-the-art attribute-completion methods.

Keywords: Heterogeneous graph, graph embedding, missing attribute, neighborhood aggregation, attribute completion

1 INTRODUCTION

Heterogeneous graphs [1] are composed of multiple types of nodes and edges and can depict complex objects and their interaction in the real world, such as social networks and citation networks [2]. Particularly, the attributes of the nodes express complex information from corresponding entities [3]. However, the non-Euclidean nature of graphs makes them difficult to be analyzed by traditional machine learning models, such as convolutional neural networks [4]. Therefore, in order to mine information from graphs deeply, many graph network embedding learning models have been proposed [5]. Those models can encode the complex information of graphs into low-dimensional Euclidean space and use them as the input of other machine learning models.

Early heterogeneous graph embedding learning methods construct node sequences by random walks and transform the graph into a computable representation, such as metapath2vec [6], HIN2Vec [7] and HeteSpaceyWalk [8]. However, those methods ignore the important information contained in nodes' attributes, which makes their performance poor on heterogeneous graphs with rich node characteristics. With the in-depth study of the heterogeneous graphs, people realized that the nodes' attributes [9] express their real and rich complex connotation, and are beneficial to dig up heterogeneous graphs' information. Therefore, the current heterogeneous graph embedding learning focuses on the aggregation algorithms of nodes' attributes and proposes many excellent methods, such as HAN [10], MAGN [11] and HGT [12]. On the other hand, these models are very sensitive to the attributes of nodes. And attributes in data sets are not always complete. For example, when people collect ACM dataset [10], many nodes are non-attribute. Furthermore, existing research [13] reveals that using these models on a graph with non-attribute nodes will spread the noise in the aggregation process and affect the subsequent tasks.

To reduce the impact of non-attribute nodes on heterogeneous graph embedding learning, one common method in existing embedding learning models is to delete these non-attribute nodes at the data preprocessing stage to ensure that all of the input nodes have attributes. Nevertheless, these methods are only applicable when the total number of the non-attribute nodes is small and the attribute nodes of the same types account for a large proportion of the total dataset. In addition, this way changes the original structures of graphs, and cannot analyze non-attribute nodes. Another common method is to fill the attributes of non-attribute nodes with zero vectors. However, this way also causes noise propagation in the graphs after attribute aggregation of nodes, resulting in poor tasks. The latest research on attribute completion which is called HGN [13] uses the first-order neighbors of

non-attributive nodes to complete attributes. This method only considers that some types of nodes are non-attributive, so it is necessary to ensure that the first-order neighbor classes of these non-attribute class nodes are complete. Yet, non-attribute nodes are likely to occur randomly in each category, still using this method will lead to poor performance of downstream tasks. Consequently, how to accurately complete non-attribute nodes in a heterogeneous graph is an important problem that needs to be studied urgently in the current heterogeneous graph embedding learning.

For solving the shortcomings of existing methods in dealing with the data sets with missing attributes, we propose an attribute completion method of heterogeneous graph based on attribute neighborhood (HGCAN). This method solves the above problems by applying two stages of attribute completion.

Specifically, in the first stage of attribute completion, we first use an attribute neighborhood selection strategy which is based on a meta-path to construct attribute neighborhoods of non-attribute nodes. Therefore, all the nodes in the generated attribute neighborhood have attributes. Then, a graph attention mechanism is used to obtain the weight of the attribute nodes to the non-attribute nodes in each attribute neighborhood. Through weighted aggregation, the non-attribute nodes are initially completed. In the second stage of attribute completion, We fuse the original topological embedding of non-attribute nodes with their initial completion attributes. Topological embedding contains the distance relations between nodes. In this way, the completed attributes of nodes obtain rich information, and can be well combined with other heterogeneous graph embedding learning models to make the whole system end-to-end. Note that we use the loss function of the attribute completion and the loss function of the model training task as the final loss to optimize the learning process.

The main contributions of this paper are summarized as follows:

- This paper finds a problem of missing whole attributes of some nodes in the heterogeneous graph. Previous heterogeneous graph attribute completion methods usually directly aggregate first-order neighbors. This only applies to the case that some categories of nodes are non-attributive. This paper extends the study of such issues. As far as the literature we have investigated, this is the first attempt to effectuate the attribute completion of a heterogeneous graph when the non-attribute nodes occur in each type.
- This paper proposes a method of a heterogeneous graph attribute completion based on the attribute neighborhood. When there are non-attribute nodes in each node type, this framework makes up for the shortcomings of previous methods in dealing with missing attributes, and is easy to combine with heterogeneous graph embedding learning models.
- In this paper, a large number of experiments have been conducted on ACM dataset to evaluate the performance of HGCAN. The results show that the performance of the node attributes fulfilled by HGCAN in downstream tasks is always better than the most advanced attribute completion baseline model.

In addition, ablation experiments were carried out on ACM dataset to further prove the effectiveness of the proposed framework.

The remainder of this paper is arranged as follows. Section 2 presents the related work of heterogeneous graph embedding learning and attribute completion. Section 3 defines the problem. Section 4 represents the proposed mechanism in detail. Section 5 discusses the experimental results. Finally, the paper is summarized in Section 6.

2 RELATED WORK

The goal of graph embedding [14] is to map the nodes in the graphs into low-dimensional vector representations [15] and retain as much original information of the graph as possible in Euclidean space for application in downstream tasks. Early graph embedding methods were first applied to homogeneous graphs. For example, Skip Gram's model generalization [16] and line [17], these methods can well capture the topological structure of the graph and reduce the impact of sparsity. However, these methods are not designed to deal with node attributes, and it is difficult to perform well for most graphs with rich node attributes. With the development of neural networks, there are spectral-based (such as GCN [3] and AGCN [18], which use edge information to generate new node representations) and spatial-based graph neural network models (such as GraphSage [2] and GAT [19], which directly convolve in the graph domain by aggregating node information). Besides, UG-AGE [20] learns complex semantics of edges by generating fake neighbors as negative samples. GFN [21] can generate text graphs and learn different opinions in the graph reasoning stage. Although these methods are suitable for homogeneous graphs and depend on complete node attributes, they provide many ideas for heterogeneous graph embedding learning.

Different from homogeneous graphs, heterogeneous graphs usually need to consider the dissimilarity of neighbor information under various relationships. Inspired by the learning of homogeneous graphs, most heterogeneous graphs deal with the heterogeneity of graphs through meta-paths. For example, metapath2vec [6] generates a random traversal guided by a single meta-path and then feeds it to the skip-gram model to engender node embedding. However, such methods do not utilize the attributes of the nodes. HAN [10] transforms a heterogeneous graph into several homogeneous graphs based on multiple meta-paths and employs an attention mechanism to aggregate diverse meta-paths. MAGN [11] expands HAN [10] by considering both the messages of multiple meta-paths and intermediate semantic nodes on the meta-paths. DT-GCN [22] embeds nodes in hyperbolic space. BM-GCN [23] introduces blocking modeling to GCN and learns the corresponding aggregation rules for neighbors of different classes. P-GCN [24] proposes two measurements of homophily degree, which can constrain the similarity of representations between nodes, to adaptively learn the propagation process. TUD-GSL [25] proposes a SUBLIME framework to learn the structures of graphs. Besides, RICE [26] proposes

a balanced sampling strategy guided Contrastive Learning mechanism to deal with the long-tail problem and the incremental graph learning on social information. These methods are proficient to capture the complex details in the heterogeneous graphs. The learned low-dimension embeddings execute properly in several downstream tasks such as node classification and link prediction, but still rely on the heterogeneous graphs with full attributes.

In view of the fact that more and more graph embedding learning methods are highly sensitive to node attributes, non-attribute nodes will reduce the effectiveness of these models. Therefore, to ensure the integrity of the attribute set, we can learn from the relevant attribute completion methods in machine learning [27]. For instance, MacKay [28] proposed Bayesian interpolation method in 1992. Rodriguez et al. [29] suggested an improved method for the single heat vector coding in 2018. Han et al. [30] interpolated the average value of traffic flow data. Yet, these methods make the attribute completion of nodes and graph embedding learning become two independent parts. Some research [13] shows that the completed attributes in the preprocessing stage are likely to bring negative effects for embedding learning.

With the development of deep learning, more and more people combine attribute completion with model learning. Based on the idea of Generate Adversarial Network(GAN), Yoon et al. [31] proposed the GAIN method to interpolate medical data, and Jiang et al. [32] also used GAN to interpolate sensor missing data for long-term detection of bridge health. Lall and Robinson [33] yielded a multiple interpolation method with a denoising automatic encoder, which improved the interpolation efficiency for large-scale datasets. However, the above methods are used for Euclidean structure data and cannot handle the non-European structure of graphs. Recently, the HGN method proposed by Jin et al. [13] considers that when all the attributes of some types of nodes are missing, the attributes of other types of nodes directly connected can be used to complete them, instead of using one-hot vectors.

In general, the existing heterogeneous graph embedding learning methods depend on a complete set of node attributes. However, the existing attribute completion methods still have some limitations. Manual attribute completion cannot determine the accuracy of attribute completion. Therefore, this paper proposes a heterogeneous graph attribute completion method based on attribute neighborhood to complete node attributes efficiently and accurately.

3 PROBLEM STATEMENT

This section first provides formal definitions of some significant terminologies related to heterogeneous graphs used throughout this paper and then formalizes the problem.

Definition 1 (Attribute Heterogeneous Graph)**.** Heterogeneous graph G is defined as a 5-tuple $G = (V, E, T_V, T_E, X)$ where V is the node set, E represents the set of edges, T_V and T_E indicate the sets of node and edge type, respectively, with $T_V + T_E > 2$. Each node $v \in V$ is associated with its mapping function $\varphi(v) : V \rightarrow$

T_V, while each edge $e \in E$ is accompanied with the mapping function $\varphi(e) : E \to T_E$. The X represents the collection of nodes' attributes.

Definition 2 (Attribute Missing Heterogeneous Graph)**.** Given an attribute heterogeneous graph $G = (V, E, T_V, T_E, X)$, missing attribute means that for the array of nodes' attributes X, $\exists v \in V$, node v has no attribute. We mark the missing attribute with a zero vector of the same dimension. We take V_{ZERO} to indicate the nodes in the node set whose attributes are zero vectors, so there is $V_{ZERO} \in V$ and $X_{ZERO} \neq \emptyset$. In addition, non-attribute nodes are not of the same type. That is, when we use node type T_{VZERO} to contain the type of nodes in V_{ZERO}, then there is $T_{VZERO} = T_V$.

As shown in Figure 1, ACM dataset can be composed of three types of nodes: paper (P), author (A) and subject (S), and two types of edges: author-paper and subject-paper. Each type has some non-attribute nodes. In this paper, zero vectors are used to identify the non-attribute nodes.

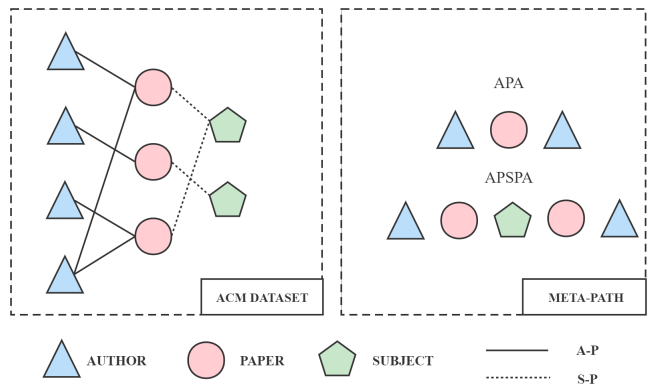


Figure 1. ACM dataset and its meta-paths

Definition 3 (Meta-path)**.** A meta-path P is defined as a path in the form of $T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} T_{l+1}$, which describes a composite relationship between node type T_1 and T_{l+1}, $R = R_1 \circ R_2 \circ \ldots R_l$, where $\circ$ represents a composite operator on the relationship.

Taking Figure 1 as an example, the meta-path "APA" describes the relationship between two authors (A) on one paper (P). The meta-path "APSPA" represents that two authors publish articles (P) on the same subject (S). Recent works show that many data mining tasks in heterogeneous graphs can benefit from the meta-path modeling.

As is shown in Figure 2, there may be non-attributive nodes in different types of nodes, which will have a negative influence on heterogeneous graph embedding

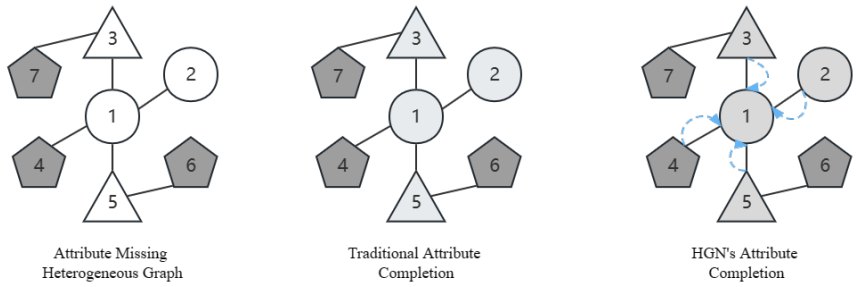


Figure 2. ACM dataset with missing attributes and its attribute completion methods

learning. What is more, the existing attribute completion methods have certain limitations. Traditional completion methods separate attribute completion and embedding learning, which make the completed attributes unreliable. However, HGN [13], which is the latest advanced heterogeneous graph attribute completion relies on the first-order neighbors' attributes. It may lead to the deviation of the completed attributes.

Therefore, this paper further develops the research on the completion method of missing attributes of heterogeneous graphs and formalizes the research questions as follows.

Question 1 (Attribute completion of attribute missing heterogeneous graphs)**.** Given an attribute missing heterogeneous graph $G = (V, E, T_V, T_E, X)$, in which some nodes have no attributes, how to learn a d-dimension node representation $x \in R^{|V|\times d}$, where $d = |V|$, so that the learned node representation x can effectively replace the non-attribute nodes.

4 HGCAN FRAMEWORK

This section mainly introduces the framework of a heterogeneous graph completion method based on attribute neighborhood (HGCAN). HGCAN can effectively solve the problem of poor embedding learning caused by the lack of some attributes of the heterogeneous graph.

4.1 Overview

To complete the non-attributive nodes, the main idea of HGCAN is that the attributes of non-attributive nodes are completed by the attributes of the nodes which have relationships with non-attributive nodes. In order to capture the attribute nodes with semantic relationships, we first find the attribute nodes with semantic relationships, and then learn the semantic relationships among the distance between nodes through structural information. Therefore, HGCAN is mainly composed of

two stages of attribute completion, which is shown in Figure 3. To be specific, given an attribute missing heterogeneous graph $G = (V, E, T_V, T_E, X)$ with some non-attribute nodes, HGCAN first uses the existing topology embedding algorithm [6] to pre-learn the topological embedding matrix s of nodes. Some reports [23, 24] show that the topological embedding of a node can reflect the semantic relationship between this node and other nodes. Next, HGCAN applies two stages of the attribute completion.

In the first stage of the attribute completion, HGCAN exercises the attribute neighborhood selection strategy guided by meta-paths to capture the attribute nodes which have semantic relationships with the non-attribute node and generate the attribute neighborhoods of each non-attribute node. Then, HGCAN implements the graph attention mechanism guided by s on each attribute neighborhood, and learns the weights of nodes with attributes. By aggregating the attributes of neighborhood nodes, HGCAN obtains the preliminary completion embedding X_C of the node. In the second stage of attribute completion, HGCAN learns the near and far relationship between nodes by fusing the topological embedding and the initial completion attributes of the non-attribute nodes. The farther away from the non-attributive node, the less influence the node with attributes has on it. After two stages of the attribute completion, we acquire the node embeddings.

The obtained node embeddings can be combined with any heterogeneous graph embedding model. In a word, we send the complete node attributes and the original heterogeneous graph G into the model as a new graph for learning. Then we use the attribute completion loss function and task loss function to optimize the whole model end-to-end.

The following sections will introduce the content of the framework in detail.

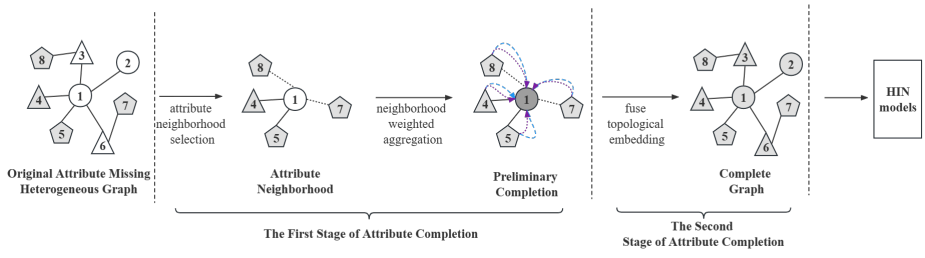


Figure 3. Attribute completion framework of heterogeneous graphs based on attribute neighborhood

4.2 The First Stage of Attribute Completion

To complete non-attribute nodes, some studies average the directly connected neighbors. The state-of-the-art study weights and aggregates first-order neighbors. Note that these methods only focus on first-order neighbors, but first-order neighbors may also have non-attribute nodes. Therefore, we propose the first stage of attribute completion, which is composed of two modules: Attribute Neighborhood Selection and Neighborhood Weighted Aggregation. The core is the attribute neighborhood, which can capture the semantic characteristics of attribute nodes versus non-attribute nodes. More specifically, in the first stage of attribute completion, we first use the Attribute Neighborhood Selection module, which is a strategy based on meta-path. Meta-path is an important concept in heterogeneous graphs, which reflects the high-level semantic relationship of heterogeneous graphs. For example, the meta-path "APA" reflects the co-author relationship between the two authors. Therefore, we use meta-path to construct attribute node sequences through a random walk, which is called attribute neighborhood. Attribute neighborhood contains attribute nodes which have semantic relationships with non-attribute nodes. We construct it and then we aggregate it to preliminary completed non-attribute nodes. In particular, Neighborhood Weighted Aggregation uses a graph attention mechanism to focus on the topological structure similarity of nodes, so as to learn the importance between attribute nodes and non-attribute nodes. We get preliminary attributes by weighted aggregation, which contains the semantic relationships from attribute nodes to non-attribute nodes.

4.2.1 Attribute Neighborhood Selection

The existing models work out the problem that some kinds of nodes have no attributes by weighted averaging the attributes of neighbor nodes directly connected. However, considering that the non-attribute nodes may exist in distinct categories, straightly connected neighbor nodes do not necessarily have attributes. If we only use zero vectors instead of them, it will make a deviation in attribute aggregation. Therefore, the first stage of attribute completion proposes an attribute neighborhood selection strategy based on the meta-path to solve this, which is, the meta-paths are used to generate a sequence of multiple types of nodes, then the node sequences are used to find the attribute neighbors of the non-attribute nodes on the meta-paths and these attribute nodes are regarded as the attribute neighborhood of the node.

Formally, given a heterogeneous graph $G = (V, E, T_V, T_E, X)$ and multiple manually selected meta-path patterns $P_j = T_1 \overset{R_1}{\rightarrow} T_2 \overset{R_2}{\rightarrow} \dots T_t \overset{R_t}{\rightarrow} T_{t+1} \dots \overset{R_l}{\rightarrow} T_{l+1}$ as the guidance of random walk, where $j \in [1, J)$ means there are J meta-paths. We set the number of first-order neighbors of non-attribute nodes as the size of their attribute neighborhood. Therefore, the size of non-attribute nodes in the first-order neighbors is the length of a random walk sequence. The transition probability of

step i can be defined as follows:

$$p\left(v_{i+1}\middle|v_i^t, P\right) = \begin{cases} \frac{1}{|M(v_{i+1})|}, & (v_i^t, v_{i+1}) \in E, \varphi\left(v_{i+1}\right) = t+1, v_{i+1} \notin V_{zero}, \\ \frac{1}{|N(v_{i+1})|}, & (v_i^t, v_{i+1}) \in E\ , \varphi\left(v_{i+1}\right) = t+1, v_{i+1} \in V_{zero}, \\ 0, & \text{others}, \end{cases} \tag{1}$$

where $v_i^t \in T_t, v_{i+1} \in T_{t+1}$ indicates that the type of current node v_i^t is T_t and the type of next node v_{i+1} after random walk is T_{t+1}, depending on the predefined meta-path P. $M\left(v_{i+1}\right)$ represents the length of the node set where nodes in it have attributes and are connected to node v_i^t. E represents the edge set, and V_{ZERO} represents all non-attribute nodes. $N\left(v_{i+1}\right)$ represents the length of the node set where nodes in it are connected to node v_i^t but with no attributes. When the attribute of v_{i+1} is not empty, that is, $v_{i+1} \notin V_{ZERO}$, the probability of the next step is $\frac{1}{|M(v_{i+1})|}$. Otherwise, the probability of the next step is $\frac{1}{|N(v_{i+1})|}$. Generally, the type of node in the next step counts on the predefined meta-path, and the specific node banks on whether the node has attributes. When the node sequence length does not meet the predefined length, the next step is to select an attribute node and add it into the node sequence. If there is no attribute node in the next step, we select non-attribute node. However, the non-attribute nodes do not belong to the attribute neighborhood, and are only used as intermediate nodes to generate the node sequences based on the meta-path. When we find enough attribute nodes, we construct an attribute neighborhood.

For more detail, we regard Equation (1) as the probability formula for a single node to the next node. To generate attribute neighborhood, we use the nodes with attributes in the first-order neighborhood of non-attribute nodes as part of the attribute neighborhood. The remaining nodes are found by a random walk strategy. That is, we only need to perform the random walk algorithm on the non-attribute nodes in the first-order neighborhood. For J meta-paths, we build J random walk sequences for each non-attribute node, and select the node with a high probability of occurrence to complete the attribute neighborhood.

In the traditional neighborhood selection algorithms, although there are some algorithms based on the meta-path, almost all of them only consider the information of directly connected nodes and do not consider whether all nodes in the neighborhood contain attributes. For example, in Figure 2, for node 1, the traditional method will set the non-attribute nodes 2, 3, and 6 to zero vectors, which will cause noise to propagate on the graph, resulting in a poor aggregation effect. The random walk strategy adopted in this paper is based on the mode of meta-path. It tries to find the nodes with attributes based on the meta-paths. That is to say, in order to ensure that the neighborhood with attributes is obtained, the strategy encourages node 1 to make jump connections for a neighborhood selection. The sequence of node 1, node 3 and node 5 is found according to the meta-path, so that the attribute neighbors of node 1 on the second hop or a further hop are found. By applying attribute neighborhood selection module, we exclude non-attribute nodes 2, 3 and 6 from the

attribute neighborhood, and we add attribute nodes 5 and 7 to form the attribute neighborhood K of node 1. Therefore, the nodes in the attribute neighborhood generated by the meta-path have a semantic relationship with the non-attribute nodes.

4.2.2 Neighborhood Weighted Aggregation

For the problem that some nodes have no attributes, as shown in Figure 2, among the first-order neighbors of node 1, nodes 2, 3, and 6 have no attributes and belong to different categories of nodes. Direct aggregation will spread the noise brought by non-attribute nodes in the network, while attribute neighborhood avoids the appearance of non-attribute nodes. However, nodes in attribute neighborhood are in various types. What is more, due to the meta-path guided strategy, attribute nodes have different semantic relationships with non-attribute nodes. Averaging attributes ignore the difference of nodes' types and the semantic relationships between nodes. Therefore, we use graph attention mechanism to assign weights to different nodes in attribute neighborhood. And then, we use different weights to aggregate attribute neighborhood, preliminary completing the non-attribute nodes.

The graph attention mechanism is used to learn the contribution of attribute nodes in attribute neighborhoods to non-attribute nodes. As shown in Figure 3, since node 1 has no attributes, HGCAN uses the pre-learning topological embeddings s to calculate the weights. Specifically, pre-learning topological information reflect different topology of nodes. Some reports [23, 24] show that authors writing same papers may have a similar topology [13], which reflects their homophily degree. Therefore, we use a meta-path based topology embedding generation algorithm [13], and in order to avoid the information loss caused by a single meta-path, we use multiple meta-paths for topology embedding learning. We use graph attention mechanism on the topological embeddings of the attribute nodes in the attribute neighborhood. The higher the similarity of topological embedding of node, the greater its importance. When we get the contribution of different attribute nodes to the non-attribute nodes, we carry out weighted aggregation on the attribute neighborhood to get the initial complement of the non-attribute nodes.

Given a node pair (v, u), where node u is a node in the attribute neighborhood of node v, the Neighborhood Weighted Aggregation module can learn the importance a_{vu} of the attribute node u to the non-attribute node v, which means that the contribution of node u to node v can be expressed as follows:

$$e_{uv} = att\left(s_u, s_v\right), \quad u \in K_v, \tag{2}$$

where $att(\cdot)$ represents the executive attention network, K_v expresses the attribute neighborhood of node v, s_u and s_v are the topological embeddings of node u and

node v. The above formula can be expanded as follows:

$$e_{uv} = LeakyReLU\left(s_v^T W_1 s_u\right), \tag{3}$$

$$a_{vu} = softmax\left(e_{uv}\right). \tag{4}$$

In Equation (3), $LeakyReLU$ is the activation function. W_1 is the parameterized attention vector of the structure. To make the coefficient easy to calculate, we apply $softmax(\cdot)$ in Formula (4) to obtain the normalized weighting coefficient a_{uv}. By executing the attention mechanism, a_{uv} can be calculated, that is, the importance of node u to node v.

Then, the first stage of attribute completion for non-attribute nodes can be finished through following formula:

$$X_v^C = \sigma\left(\sum_{u \in K_v} a_{vu} x_u\right), \tag{5}$$

where x_u represents the attributes of node u, σ is an activation function; X_v^C indicates the preliminary completed node attributes.

HGCAN extends the attention process to multi-head attention, with a view to stabilize the learning process and reduce high variance. As shown in Figure 2, different line colors represent different attention processes. HGCAN executes D times of independent attention mechanism, and then connects together. The weighted aggregation of neighborhood attributes of node v can be rewritten as follows:

$$X_v^C = \|_d^D \, \sigma\left(\sum_{u \in N_v^+} a_{vu} x_u\right). \tag{6}$$

Note that unlike other methods, HGCAN does not use the mask mechanism to focus only on the first-order nodes which are directly connected to node v. In fact, it is precisely because HGCAN has adopted the attribute neighborhood selection algorithm that it can obtain enough attribute neighbors. The previous methods only consider the directly connected nodes, but some non-attribute nodes in the first-order will lead to a decline in the quality of attribute completion. HGCAN avoids this risk. However, the weighted aggregation on the attribute neighborhood does not take into account the distance relationship between the different attribute nodes in the attribute neighborhood with the non-attribute nodes, so it can only be used as the initial complement of the non-attribute nodes. Besides, it is natural to think whether there is a case where the attribute neighborhood selection algorithm cannot find the associated node with attributes. The next section will supplement this.

4.3 The Second Stage of Attribute Completion

HGCAN uses the first stage of attribute completion to initially complete the non-attribute nodes. This stage can capture the relationship between the attribute nodes and non-attribute nodes but omit the skipped nodes. When we aggregate the attribute neighborhood, we see them as non-attribute nodes' first-order neighbors. The difference of semantic relationship between remote nodes and near nodes to non-attribute nodes is ignored. However, the contribution of the remote nodes to the non-attribute nodes may be much smaller than that of the near nodes, according to the situation that many methods only focus on first-order neighbors. It is necessary to obtain the original distance between nodes, based on original structure. Therefore, we propose the second stage of attribute completion. Non-attribute nodes' topological embeddings are pre-learned by an existing topological embedding method based on the original structure. The embeddings reflect their original topology, which can guide the distance relationships between nodes. We fuse non-attribute nodes' topological embeddings h_v and their preliminary completion embeddings X_v^C and use them as the complement attributes of nodes.

$$X_v^C = mean\left(X_v^C + W_2 h_v\right), \tag{7}$$

where $mean(\cdot)$ indicates to average the results. $+$ means to add two embeddings. W_2 represents the learnable weight parameter matrix which can reduce the dimension of h_v to X_v^C and balance these two embeddings. In addition, for the problem mentioned at the end of last section, although this situation is unlikely to occur, once it occurs, HGCAN will regard it as a zero vector and use W_2 to align the topology vector h_v of nodes with other attribute vectors.

4.4 Combining with HIN Model

In order to verify that the completed attributes can enhance the performance of heterogeneous graph embedding learning methods, evaluate the effectiveness and learnable of HGCAN in this paper, and make the system end-to-end, the learned X_v^C will replace the zero vectors in the original X_v. At the beginning, this paper uses zero vectors to represent non-attribute nodes. HGCAN then takes the heterogeneous graph $G = (V, E, T_V, T_E, X)$ after attribute completion as a new graph and sends it into any heterogeneous graph embedding learning model for combining learning.

HGCAN defines the combining loss function, that is, the loss function of attribute completion and the loss function of model training task, to optimize the learning effect. The loss function of attribute completion is as follows:

$$\mathcal{L}_{attribute} = \frac{1}{|V_U|} \sum_{u \in V_U} f_u, \tag{8}$$

where V_U means a set of nodes. Each node $u \in V_U$ has raw attributes. We randomly select some nodes from those nodes with attributes to form V_U. Then, we drop those

nodes' attributes and use HGCAN to reconstruct them. f_u represents a metric to make sure that the reconstructed attributes are as close to the raw attributes as possible. The equation of f_u is as follows:

$$f_u = \sqrt{\left(X_u^C - X_u\right)^2}, \tag{9}$$

where node u belongs to V_U and represents the nodes with attributes. X_u^C represents the reconstructed attributes of node u after using HGCAN. X_u represents the original attributes of node u. This equation calculates the Euclidean distance.

After end-to-end learning, HGCAN uses the real value of the model, *truy*, and the predicted value, *prey*, as the input of the loss function of model training task, expressed as:

$$\mathcal{L}_{classification} = L(prey, truy), \tag{10}$$

where L represents the loss function which depends on the actual tasks. In the node classification experiments, we use cross entropy loss function to optimize the overall effect.

We define loss function coefficient α to balance these two loss functions. The loss function results of the two parts of coefficient balance are optimized and learned through back propagation under the guidance of labeled data. $\mathcal{L}_{loss}$ can be formulated as:

$$\mathcal{L}_{loss} = \alpha \mathcal{L}_{attribute} + \mathcal{L}_{classification}. \tag{11}$$

5 EXPERIMENTS

To verify the improvement of the proposed attribute completion method on heterogeneous graph embedding learning, we perform experiments on the dataset of ACM. In this section, we give the experimental scheme first. Then, we describe the dataset and baselines. At last, we analyze the experiment results.

5.1 Experimental Scheme

First, we use ACM dataset in all experiments in this paper. The distribution of non-attribute nodes in ACM is also equivalent. We take three graph embedding learning models as baselines. Then we feed the embeddings of paper nodes in ACM generated by each learning model to a linear SVM and a K-means clustering model to obtain the results of node classification and clustering, respectively. Macro-F1 Value and Micro-F1 value are used as the evaluation value of multi-classification model, and normalized mutual information (NMI) and adjusted RAND index (ARI) are used as the evaluation value of clustering model. We compare those results to prove the effectiveness of the framework proposed in this paper. Besides, we use Adam optimizer and the learning rate is 0.005.

Then, to further study the necessity of each part in the framework proposed in this paper, the same dataset is studied using different variants of the proposed model. They are the baseline method without attribute completion algorithm (MAGN), a variant that only uses the first stage of attribute completion (HGCAN1), a model which only aggregates first-order neighbors but fuses the topological embeddings (HGCAN2) and the final algorithm HGCAN.

After that, in order to study some parameters involved in the model, such as the influence of different loss function coefficients on attribute completion results, this paper conducts several node clustering experiments on them to study the changes of NMI and ARI values under different loss function coefficients, so as to prove the optimal coefficient range.

Finally, in order to verify the scope of application of the model, that is, HGCAN has different effects on attribute completion of heterogeneous graphs with different ratios of non-attribute nodes, multiple random scale node attribute loss operations are conducted for a given dataset, and node clustering experiments are conducted to study the model performance under several ratios.

5.2 Dataset

The dataset of ACM used in the experiments is a subset extracted from the website of ACM[1]. By the same extraction method as HAN [10], we construct a heterogeneous graph containing 4019 papers (P), 7 167 authors (A) and 60 subjects (S), in which the papers are divided into three categories according to the conference they have been published in. We employ the meta-path set APA, APSPA to perform experiments.

What is more, each node in the ACM has attributes, and the papers have a real label information. Therefore, to facilitate the study of the problems proposed in this paper, we construct several ACM datasets with non-attribute nodes. In detail, we first select a fixed set of random seeds to ensure that the results can be reproduced. Then, according to different random seeds, we select nodes as non-attribute nodes from each node type in the same proportion. Finally, we get a missing attribute heterogeneous graph for each random seed. Here, the datasets are divided into 10 % training sets, 10 % verification sets and 80 % test sets.

5.3 Baselines

To evaluate the performance of HGCAN, we compare it with two existing methods, i.e., GAT and MAGN and the state-of-art attribute completion method HGN.

GAT [19]: As a spatial-based graph convolutional network, GAT is comprised of attention layers which can calculate different weights to different nodes in the neighborhood. Note that we test GAT on several meta-path based homogeneous graphs and report the best results.

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

MAGN [11]: As a heterogeneous graph neural network, MAGN boosts the performance through employing three steps (Node Content Transformation, Intra-meta-path Aggregation, Inter-meta-path Aggregation).

HGN [13]: As an attribute completion method for heterogeneous graph, HGN implements node attribute completion by first-order neighborhood aggregation.

5.4 Node Classification

We conduct experiments on ACM dataset in which 30 % nodes have no attributes to compare the performance of different models on node classification tasks. To make a fair comparison and reproduce the results, we set five different random seeds to loss nodes' attributes in ACM dataset randomly, and take the average value of Macro-F1 and Micro-F1 respectively as the experimental results. We verify the effectiveness by combining the proposed completion method with MAGNN. We use a linear SVM with different training ratios from 1 % to 80 % to classify paper nodes in ACM. Because SVM is a semi-supervised model, for fair comparison, only the nodes of the test set are sent to the classifier.

Besides, the parameter settings of the baseline models remain unchanged in this subsection. For the proposed framework, we use the following settings. We expand the graph attention to multiple attention with $K = 8$ attention heads. The proportion of non-attribute nodes of each category is set to 30 %, and the loss weighting coefficient λ is set to 0.5. The input data sets of all experiments were consistent.

Training Ratio	Model			
	GAT	MAGN	HGN	**HGCAN**
80 %	79.8 %	91.8 %	91.3 %	**94.5** %
60 %	79.7 %	91.5 %	90.9 %	**94.3** %
40 %	79.7 %	90.7 %	90.1 %	**93.5** %
20 %	79.7 %	89.5 %	89.1 %	**92.7** %
10 %	79.6 %	88.8 %	88.1 %	**91.6** %
5 %	79.5 %	88.9 %	87.4 %	**90.8** %
1 %	77.5 %	86.9 %	86.4 %	**89.4** %

Table 1. Macro-F1 of node classification results on ACM

As shown in Table 1 and Table 2, the proposed method HGCAN performs better than other baselines in each training ratios of ACM dataset, which proves the stability and superiority of HGCAN. Specifically, for the heterogeneous graphs with 30 % non-attribute nodes, MAGN and GAT use zero vector instead of them, and HGN uses first-order neighbors to complete them. Compared with these methods, HGCAN improves the results by 1.7 % to 3.6 %.

On the other hand, HGN uses first-order neighbors to complete the missing attributes and is combined with MAGN. However, it is worth noting that the scores in the table show that the score of HGN is not as high as MAGN. HGN only uses

Training Ratio	Model			
	GAT	MAGN	HGN	**HGCAN**
80 %	81.1 %	92.0 %	91.4 %	**94.6** %
60 %	81.0 %	91.7 %	91.0 %	**94.4** %
40 %	81.0 %	91.0 %	90.3 %	**93.7** %
20 %	81.0 %	89.7 %	89.2 %	**92.9** %
10 %	81.0 %	89.0 %	88.2 %	**91.7** %
5 %	80.9 %	89.2 %	87.7 %	**90.9** %
1 %	79.2 %	87.6 %	86.8 %	**89.6** %

Table 2. Micro-F1 of node classification results on ACM

first-order neighbors, though some of them are non-attribute. These non-attribute neighbors will lead to poor aggregation effect. Therefore, HGN shows its limitation on attribute completion. Meanwhile, HGCAN uses two stages of attribute completion to avoid aggregating non-attribute nodes, so it has greater advantages in the face of this situation.

5.5 Node Clustering

We carry out the node clustering task on ACM dataset. And we also train the models used in the node classification experiments to generate node embeddings. We use node embeddings as the input of K-means algorithm, and use NMI and ARI as evaluation indicators. NMI and ARI can measure the similarity of clustering results. The value range is between 0 and 1. The larger the value, the closer the clustering result is to the real. We make the number of clusters in K-means the number of classes in the dataset, that is, ACM papers can be divided into three categories.

As shown in Figure 4, the performance of MAGN, HGN and HGCAN is better than that of GAT, and the performance of HGN is weaker than that of MAGN. However, HGCAN still performs best in node clustering tasks. It proves again that the method proposed in this paper can obtain better node attributes.

5.6 Ablation

In order to verify the effectiveness of each stage in the framework proposed in this paper and the rationality of parameter settings, different case studies are conducted.

5.6.1 Effectiveness of Components

In this part, different HGCAN variants are tested. This experiment uses Macro-F1 to evaluate the node classification results, so the experimental settings are the same as node classification's. The methods in this paper are combined with MAGN, so MAGN is a baseline model that does not use any attribute completion method.

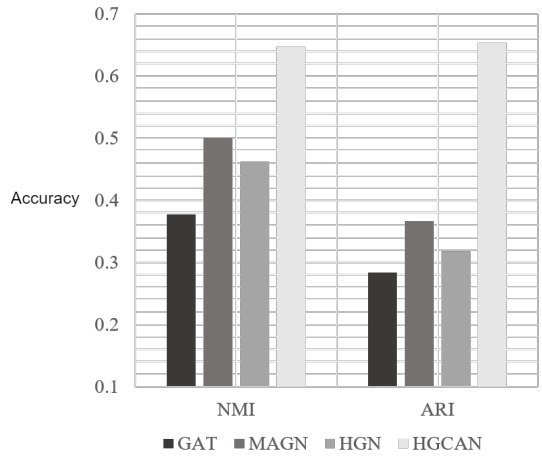


Figure 4. Node clustering results on ACM (NMI and ARI)

HGCAN1 considers selecting attribute neighborhood for aggregation, so it only uses the first stage of attribute completion. HGCAN2 considers topological embeddings to complete attributes, so it mainly uses the second stage of attribute completion. HGCAN is the final model which uses two stages of attribute completion. Except for the above differences, other parameter settings are consistent.

Training Ratio	Model			
	MAGN	HGCAN1	HGCAN2	**HGCAN**
80 %	92.0 %	92.3 %	93.2 %	**94.6** %
60 %	91.7 %	92.1 %	92.8 %	**94.4** %
40 %	91.0 %	91.3 %	92.1 %	**93.7** %
20 %	89.7 %	90.6 %	91.2 %	**92.9** %

Table 3. Effectiveness of components

As shown in Table 3, HGCAN1 achieves better performance than MAGN by applying the attribute neighborhood algorithm, which shows that the attribute neighborhood effectively reduces the noise propagation caused by missing attributes when aggregating attributes. HGCAN2 fuses the topological embeddings with the completion of first-order neighborhood aggregation. The results indicate that considering the distance relationship between nodes can promote the attribute completion of nodes, thus proving the necessity of the second stage of attribute completion. Finally, the results of HGCAN1, HGCAN2 and HGCAN show that each stage is necessary and promotive. The three methods are still better than the best baseline MAGN, which shows the effectiveness of the proposed framework in this paper.

5.6.2 Loss Function Coefficient α

To achieve the best performance of the model, this section tests different loss function coefficients which are represented by symbol α. Under the condition that other parameters are the same, we use symbol α to represent the loss function coefficient and use NMI and ARI evaluation indicators as the final scores. The results are shown in Figure 5. With the increase of the coefficient, the score shows a trend of rising first and then falling. The score proves that if the proportion of loss function of attribute completion is too high, the task effect will be affected, and if it is too small, the attribute completion effect will be affected too.

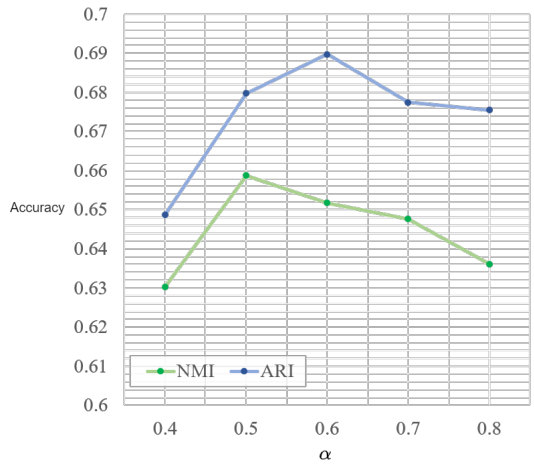


Figure 5. Loss function coefficient comparison

5.6.3 The Scale of Non-Attribute Nodes

To verify the applicability of the model, this section tests the impact of different ratios of non-attribute nodes on the performance of the proposed model. The results are shown in Figure 6. With the increase of the proportion of non-attribute nodes, the scores of NMI and ARI show a downward trend. This is reasonable, because the more attributes are missing and the fewer nodes are used for attribute completion, the greater the noise and the greater the impact on the final score. If the scale of non-attribute nodes is too high or too small, the results of other experiments may lose credibility. Therefore, in the node classification and clustering experiments in this paper, 30 % of the missing attributes are selected to complete the experiment.

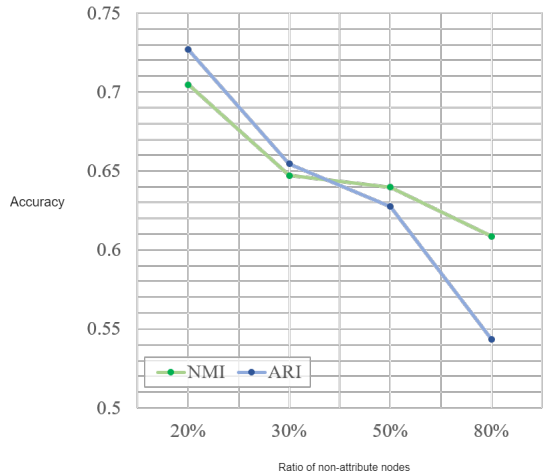


Figure 6. The scale of non-attribute nodes

6 CONCLUSIONS

In this paper, for the non-attribute nodes in the heterogeneous graph, we propose an attribute completion model based on attribute neighborhood named HGCAN. HGCAN uses two stages of attribute completion to capture rich information of nodes, making the completed node attributes accurate and effective, and improving the mining effect of the heterogeneous graph. In short, the attribute completion in the first stage uses the attribute neighborhood selection strategy and the neighborhood weighted aggregation module to capture the sequence of attribute nodes which have semantic relationships with non-attribute nodes and do the initial completion of nodes' attributes. In the second stage, the attribute completion uses the original structure information of the non-attribute node to capture the far and near semantic relationships between attribute nodes and the non-attribute node in the structure. The secondary completion of nodes' attributes is obtained by fusing the topological embedding and the initial completion of non-attribute nodes. Finally, we send the completed heterogeneous graphs into the existing heterogeneous graph embedding models for joint learning. In experiments, we use MAGNN model as the combined model. The results on node classification, node clustering and ablation experiments demonstrate the superiority of HGCAN over the existing state-of-the-art works convincingly.

There are several potential improvements and extensions to the attribute completion method proposed in this paper that could be addressed as future work, such as overcoming the dependency of attribute neighborhood on prior knowledge like a meta-path and the impact of noise from the original attributes on the method. What is more, it is also beneficial to extend the method to fill in heterogeneous

graphs with different attribute missing ratios, which will allow us to tackle more problems of missing attributes.

Acknowledgment

This work was supported by the Shanghai Science and Technology Innovation Action Plan Project (No. 22511100700).

REFERENCES

[1] Shi, C.—Li, Y.—Zhang, J.—Sun, Y.—Philip, S. Y.: A Survey of Heterogeneous Information Network Analysis. IEEE Transactions on Knowledge and Data Engineering, Vol. 29, 2016, No. 1, pp. 17–37, doi: 10.1109/TKDE.2016.2598561.

[2] Hamilton, W.—Ying, R.—Leskovec, J.: Inductive Representation Learning on Large Graphs. In: Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 30 (NIPS 2017). Curran Associates, Inc., 2017, pp. 1024–1034, doi: 10.48550/arXiv.1706.02216.

[3] Kipf, T. N.—Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. 5th International Conference on Learning Representations (ICLR 2017), 2016, doi: 10.48550/arXiv.1609.02907.

[4] Yu, S.—Lee, D.—Yu, H.: Convolutional Neural Networks with Compression Complexity Pooling for Out-of-Distribution Image Detection. Proceedings of the Twenty-Ninth International Conference on Artificial Intelligence (IJCAI-20), 2020, pp. 2435–2441, doi: 10.24963/ijcai.2020/337.

[5] Wang, X.—Bo, D.—Shi, C.—Fan, S.—Ye, Y.—Yu, P. S.: A Survey on Heterogeneous Graph Embedding: Methods, Techniques, Applications and Sources. IEEE Transactions on Big Data, Vol. 9, 2023, No. 2, pp. 415–436, doi: 10.1109/TBDATA.2022.3177455.

[6] Dong, Y.—Chawla, N. V.—Swami, A.: metapath2vec: Scalable Representation Learning for Heterogeneous Networks. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17), 2017, pp. 135–144, doi: 10.1145/3097983.3098036.

[7] Fu, T. Y.—Lee, W. C.—Lei, Z.: HIN2Vec: Explore Meta-Paths in Heterogeneous Information Networks for Representation Learning. Proceedings of the 2017 ACM Conference on Information and Knowledge Management (CIKM '17), 2017, pp. 1797–1806, doi: 10.1145/3132847.3132953.

[8] He, Y.—Song, Y.—Li, J.—Ji, C.—Peng, J.—Peng, H.: HeteSpaceyWalk: A Heterogeneous Spacey Random Walk for Heterogeneous Information Network Embedding. Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19), 2019, pp. 639–648, doi: 10.1145/3357384.3358061.

[9] Berahmand, K.—Nasiri, E.—Pir Mohammadiani, R.—Li, Y.: Spectral Clustering on Protein-Protein Interaction Networks via Constructing Affinity Matrix Us-

ing Attributed Graph Embedding. Computers in Biology and Medicine, Vol. 138, 2021, Art. No. 104933, doi: 10.1016/j.compbiomed.2021.104933.

[10] Wang, X.—Ji, H.—Shi, C.—Wang, B.—Ye, Y.—Cui, P.—Yu, P. S.: Heterogeneous Graph Attention Network. The World Wide Web Conference (WWW '19), 2019, pp. 2022–2032, doi: 10.1145/3308558.3313562.

[11] Fu, X.—Zhang, J.—Meng, Z.—King, I.: MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. Proceedings of The Web Conference 2020 (WWW '20), 2020, pp. 2331–2341, doi: 10.1145/3366423.3380297.

[12] Hu, Z.—Dong, Y.—Wang, K.—Sun, Y.: Heterogeneous Graph Transformer. Proceedings of The Web Conference 2020 (WWW '20), 2020, pp. 2704–2710, doi: 10.1145/3366423.3380027.

[13] Jin, D.—Huo, C.—Liang, C.—Yang, L.: Heterogeneous Graph Neural Network via Attribute Completion. Proceedings of The Web Conference 2021 (WWW '21), 2021, pp. 391–400, doi: 10.1145/3442381.3449914.

[14] Liu, Y.—Jin, M.—Pan, S.—Zhou, C.—Zheng, Y.—Xia, F.—Yu, P. S.: Graph Self-Supervised Learning: A Survey. IEEE Transactions on Knowledge and Data Engineering, Vol. 35, 2023, No. 6, pp. 5879–5900, doi: 10.1109/TKDE.2022.3172903.

[15] Cui, P.—Wang, X.—Pei, J.—Zhu, W.: A Survey on Network Embedding. IEEE Transactions on Knowledge and Data Engineering, Vol. 31, 2019, No. 5, pp. 833–852, doi: 10.1109/TKDE.2018.2849727.

[16] Mikolov, T.—Chen, K.—Corrado, G.—Dean, J.: Efficient Estimation of Word Representations in Vector Space. CoRR, 2013, doi: 10.48550/arXiv.1301.3781.

[17] Tang, J.—Qu, M.—Wang, M.—Zhang, M.—Yan, J.—Mei, Q.: LINE: Large-Scale Information Network Embedding. Proceedings of the 24th International Conference on World Wide Web (WWW '15), 2015, pp. 1067–1077, doi: 10.1145/2736277.2741093.

[18] Peng, Z.—Liu, H.—Jia, Y.—Hou, J.: Attention-Driven Graph Clustering Network. Proceedings of the 29th ACM International Conference on Multimedia (MM '21), 2021, pp. 935–943, doi: 10.1145/3474085.3475276.

[19] Veličković, P.—Cucurull, G.—Casanova, A.—Romero, A.—Lio, P.—Bengio, Y.: Graph Attention Networks. CoRR, 2017, doi: 10.48550/arXiv.1710.10903.

[20] Li, J.—Fu, X.—Zhu, S.—Peng, H.—Wang, S.—Sun, Q.—Yu, P. S.—He, L.: A Robust and Generalized Framework for Adversarial Graph Embedding. IEEE Transactions on Knowledge and Data Engineering, Vol. 35, 2023, No. 11, pp. 11004–11018, doi: 10.1109/TKDE.2023.3235944.

[21] Dai, Y.—Shou, L.—Gong, M.—Xia, X.—Kang, Z.—Xu, Z.—Jiang, D.: Graph Fusion Network for Text Classification. Knowledge-Based Systems, Vol. 236, 2022, Art. No. 107659, doi: 10.1016/j.knosys.2021.107659.

[22] Shen, Y.—Li, Z.—Wang, X.—Li, J.—Zhang, X.: DataType-Aware Knowledge Graph Representation Learning in Hyperbolic Space. Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), 2021, pp. 1630–1639, doi: 10.1145/3459637.3482421.

[23] He, D.—Liang, C.—Liu, H.—Wen, M.—Jiao, P.—Feng, Z.: Block Modeling-Guided Graph Convolutional Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, No. 4, pp. 4022–4029, doi: 10.1609/aaai.v36i4.20319.

[24] Wang, T.—Jin, D.—Wang, R.—He, D.—Huang, Y.: Powerful Graph Convolutional Networks with Adaptive Propagation Mechanism for Homophily and Heterophily. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, No. 4, pp. 4210–4218, doi: 10.1609/aaai.v36i4.20340.

[25] Liu, Y.—Zheng, Y.—Zhang, D.—Chen, H.—Peng, H.—Pan, S.: Towards Unsupervised Deep Graph Structure Learning. Proceedings of the ACM Web Conference 2022 (WWW '22), 2022, pp. 1392–1403, doi: 10.1145/3485447.3512186.

[26] Peng, H.—Zhang, R.—Li, S.—Cao, Y.—Pan, S.—Yu, P. S.: Reinforced, Incremental and Cross-Lingual Event Detection from Social Messages. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 45, 2022, No. 1, pp. 980–998, doi: 10.1109/TPAMI.2022.3144993.

[27] Deng, J.—Lubao, S.—Deqiang, H.—Rui, T.: Processing Method of Missing Data and Its Developing Tendency. Statistics and Decision, Vol. 35, 2019, No. 23, pp. 28–34, doi: 10.13546/j.cnki.tjyjc.2019.23.005 (in Chinese).

[28] MacKay, D. J. C.: Bayesian Interpolation. Neural Computation, Vol. 4, 1992, No. 3, pp. 415–447, doi: 10.1162/neco.1992.4.3.415.

[29] Rodríguez, P.—Bautista, M. A.—Gonzalez, J.—Escalera, S.: Beyond One-Hot Encoding: Lower Dimensional Target Embedding. Image and Vision Computing, Vol. 75, 2018, pp. 21–31, doi: 10.1016/j.imavis.2018.04.004.

[30] Han, W.—Wang, J.—Hu, J.: Interpolation Method for Missing Values in Traffic Flow Data. Transportation and Computers, Vol. 23, 2005, No. 1, pp. 39–42, doi: 10.3963/j.issn.1674-4861.2005.01.010 (in Chinese).

[31] Yoon, J.—Jordon, J.—Schaar, M.: GAIN: Missing Data Imputation Using Generative Adversarial Nets. In: Dy, J., Krause, A. (Eds.): Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research (PMLR), Vol. 80, 2018, pp. 5689–5698, http://proceedings.mlr.press/v80/yoon18a/yoon18a.pdf.

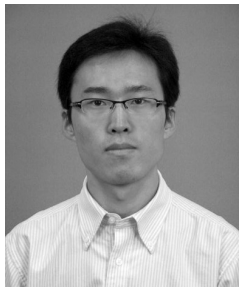
[32] Jiang, H.—Wan, C.—Yang, K.—Ding, Y.—Xue, S.: Continuous Missing Data Imputation with Incomplete Dataset by Generative Adversarial Networks-Based Unsupervised Learning for Long-Term Bridge Health Monitoring. Structural Health Monitoring, Vol. 21, 2022, No. 3, pp. 1093–1109, doi: 10.1177/14759217211021942.

[33] Lall, R.—Robinson, T.: The MIDAS Touch: Accurate and Scalable Missing-Data Imputation with Deep Learning. Political Analysis, Vol. 30, 2022, No. 2, pp. 179–196, doi: 10.1017/pan.2020.49.

Zhaohui Zhang received his B.Sc. degree in computer science from the Anhui Normal University, Wuhu, China in 1994. He obtained his Ph.D. degree in computer science from the Tongji University, Shanghai, China in 2007. From 1994 to 2015, he worked in the Anhui Normal University as Professor. Since 2015 he has been working as Professor in the School of Computer Science and Technology, Donghua University, Shanghai, China. His research interests include big data intelligent processing and behavior analysis.

Siting Huang received her B.Sc. degree in computer science and technology from the Donghua University, Shanghai, China, in 2020, and now she is pursuing her M.Sc. degree in computer science and technology from the Donghua University, Shanghai, China. Her current research interests include graph embedding, machine learning and attribute completion.

Chaochao Hu received his B.Sc. degree in computer science and technology from the Shangrao Normal University, Shangrao, China, in 2019, his M.Sc. degree in software engineering from the Henan Polytechnic University, Jiaozuo, China, in 2022, and now he is pursuing Ph.D. degree in enterprise information system and engineering from the Donghua University, Shanghai, China. His current research interest are graph neural networks.



Pengwei Wang received his B.Sc. and M.Sc. degrees from the Shandong University of Science and Technology, Qingdao, China, in 2005 and 2008, respectively, and his Ph.D. degree from the Tongji University, Shanghai, China, in 2013, all in computer science. He finished his postdoctoral research work at the Department of Computer Science, University of Pisa, Italy, in 2015. He is currently serving as Associate Professor in the School of Computer Science and Technology, Donghua University, Shanghai. His research interests include cloud computing, data mining, and service computing.