Expert Systems With Applications 243 (2024) 122945

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

RA-HGNN: Attribute completion of heterogeneous graph neural networks based on residual attention mechanism

Zongxing Zhao, Zhaowei Liu *, Yingjie Wang, Dong Yang, Weishuai Che

Yantai University, School of Computer and Control Engineering, YanTai, 264005, ShanDong, China

A R T I C L E I N F O

Keywords:
Graph neural networks
Heterogeneous information networks
Residual attention mechanism
Attribute completion

A B S T R A C T

Heterogeneous graphs, which are also called heterogeneous information networks, analyze the different types of nodes in an information network and the different types of links between them to accurately tell the difference between different semantics. In recent years, there have been several GNN-based models to process heterogeneous graph data and achieve good performance. The model faces the challenge of first considering how to deal with the challenges posed by embedding different types of nodes in a heterogeneous graph; secondly, analyzing the node attribute information, which requires satisfying all nodes with attributes, which is not easy to achieve due to the existence of individual nodes and their neighbors that do not carry attributes. Previous network structures have added attributes to nodes by handcrafted methods, thus neglecting the overall learnability of the model, which in turn leads to poor performance. This paper analyzes the reasons for this phenomenon and aims to design a learning-competent heterogeneous graph neural networks(HGNN) framework. The understanding in this study embeds different types of nodes into the same feature space for node embedding, using the topological embedding of heterogeneous graphs as a guide to complete the process of complementing non-attributed nodes through learnable ways in the model and the use of residual attention mechanisms to handle attributes between nodes. Therefore, this paper proposes a general framework for Attribute Completion of Heterogeneous Graph Neural Network Based on Residual Attention Mechanism (RA-HGNN) , and combines it with other GNN models to enable end-to-end execution of the entire model. Experimental verification was completed on real-world data sets to prove the feasibility of the model, and the experimental results showed state-of-the-art performance.

1. Introduction

In our natural world, there are many complex systems, such as social networks, shopping systems, and documentation systems, which graph data structures can represent. In recent years, graph neural networks (GNNs) (Liu et al., 2021; Liu, Yang, Wang, Lu, & Li, 2023; Liu, Yang, Wang, & Su, 2022; Veličković et al., 2018) have generated much interest. They have become a popular direction in neural networks, but they are initially proposed for use in processing graph data. They have excellent performance in applications such as node classification, image classification and connection prediction. Especially the heterogeneous graph composed of multiple types of nodes and edges can better describe the real system, as shown in the Fig. 1(a), the IMDB can be regarded as a heterogeneous graph in which there are three types of Nodes (director, movie and actor) and two types of edges (movie-director, and movie-actor). The advantage of heterogeneous graphs is that they can include richer semantic information, while homogeneous graphs have certain limitations in this respect, which has caused scholars to study heterogeneous networks.

In recent years, heterogeneous graphs have been gradually attracting attention, and more and more heterogeneous models have been proposed. For example, HetG (Zhang, Song, Huang, Swami, & Chawla, 2019) is to transform graph data into sequence data, using Bi-LSTM (Liu et al., 2017) to capture feature interaction information. For different types of nodes, the same type of aggregation or different types of aggregation are used to complete the embedding information of nodes, and finally, the whole model is optimized by loss function as well as back propagation. MAGNN (Fu, Zhang, Meng, & King, 2020) is an analysis of heterogeneous graph neural networks based around artificially defined meta-paths (Sun, Han, Yan, Yu, & Wu, 2011), performing aggregation within and between meta-paths, mainly addressing the problem of embedding learning of heterogeneous graphs. Currently, existing embedding methods applied in heterogeneous networks are relatively

* Corresponding author.
E-mail addresses: zhaozx_ytu@163.com (Z. Zhao), lzw@ytu.edu.cn (Z. Liu), wangyingjie@ytu.edu.cn (Y. Wang), yangdong@s.ytu.edu.cn (D. Yang), cheshuaishuai@163.com (W. Che).

https://doi.org/10.1016/j.eswa.2023.122945
Received 15 February 2023; Received in revised form 12 December 2023; Accepted 12 December 2023
Available online 15 December 2023

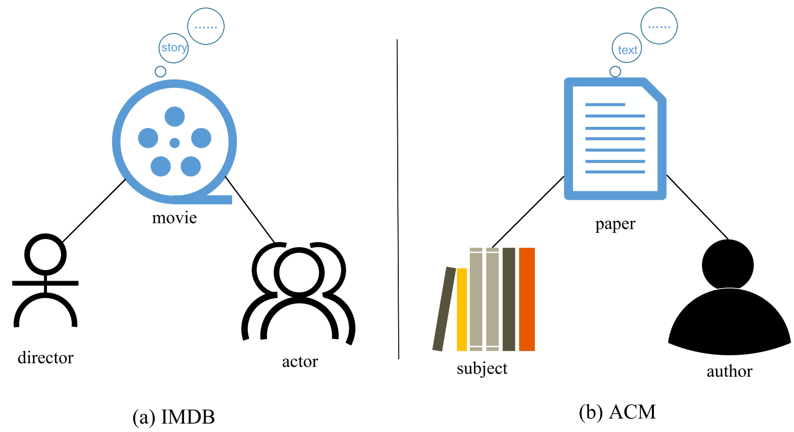


Fig. 1. This is the network structure of the IMDB dataset and the ACM dataset. Only the movie node in the IMDB dataset and the paper node in the ACM dataset have the original attributes.

limited as well as using artificial ways to define the meta-path, which inevitably results in information loss. The nature of heterogeneous nodes is the existence of multi-dimensional features. In this paper, a type conversion matrix is used for a same-dimensional node mapping, and existing methods are used for node embedding. This reduces the complexity of embedding and makes sure that the information is correct.

In deep learning (LeCun, Bengio, & Hinton, 2015), the attention mechanism (Choromanski et al., 2021; Guan, Wang, & Zhu, 2021; Liu et al., 2022; Wu et al., 2021; Xiang et al., 2022) plays an essential role in neural networks. From homogeneous graphs in the beginning to heterogeneous graphs now, the attention mechanism has been used to aggregate information about node attributes and has had the effect that was wanted in HGNN. For example, in HAN (Wang et al., 2019), the node and semantic levels are considered using the attention mechanism to weight the fusion of neighborhood and semantic information, which allows for a more comprehensive representation of node information and shows a stronger representation of the learned node representation. HRAN (Li, Liu et al., 2022) is a new heterogeneous GNN framework based on an attention mechanism that not only aggregates entity attributes from different semantic aspects but also assigns them appropriate weights to capture complex structures and rich semantics.

In most neural networks, operations are performed on the nodes in the graph structure, so information about the attributes (Xiang et al., 2020) of the nodes is essential. In performing task analysis, ideally the nodes would all be able to carry the original attributes, but in practice not all nodes are present with attributes. What is needed is to have as many contributing nodes as possible in the graph, therefore adding attributes to nodes that are non-attributes of high importance. For the problem of completing attributes, this study needs to classify the nodes in the graph structure into two categories. One is that the nodes to be considered do not have attribute information, and since the number of non-attributes is too large, only nodes of higher importance can be complemented, such as the author in Fig. 1(b); the other is that attribute information does not exist in the nodes that are not analyzed, such as actor in Fig. 1(a). IMDB has three different types of nodes, but only the movie attributes come from the bag-of-words of the plot. Also in ACM, only the paper node has attributes directly from its key. The specific explanation is that in ACM, the author node is usually analyzed for task processing, but the corresponding key attributes are lacking. In IMDB, task processing is usually performed on movie nodes with attributes, but experiments show that the lack of attributes on director and actor nodes will also affect the effect of tasks.

In heterogeneous information networks, for nodes with missing attributes, there will be neighbor nodes with attributes. Studying neighbor nodes with attributes to complete attribute completion work for nodes with missing attributes is something that existing heterogeneous networks lack. This paper proposes a framework for attribute completion of heterogeneous graph neural network based on residual attention mechanism to achieve feature conversion and cross-type information transfer of heterogeneous data, which is more conducive to model training and learning. Specifically, in the node embedding, a type conversion matrix is introduced, which can map the nodes in the heterogeneous network to the same feature space, aiming to optimize the heterogeneous network graph embedding problem, solving the problem of difficulty in dealing with multi-dimensional node features in traditional graph neural networks (Luan et al., 2022; Park, Rossi, Ahmed, & Faloutsos, 2022; Zhang, Luo, Wang, & He, 2022), and facilitating the learnability of the heterogeneous network to ensure the accuracy of the node information. The resulting topological network then uses the residual attention mechanism to aggregate the attribute node information and perform attribute complementation for nodes with missing attributes, which not only effectively increases the number of training layers, but also solves the problems of over-smoothing, vanishing gradient of training, and missing attributes (Li, Ni et al., 2022; Lu, Li, & Wei, 2022; Yan et al., 2023) in the network of existing heterogeneous models (Xu, Dai, Zhang, & Wang, 2022; Yang, Yan, Pan, Ye, & Fan, 2023). This processing mechanism allows the framework proposed in this paper to be combined with many GNN-based heterogeneous models, and optimize the model learning process by combining the predictive loss of the model as well as the weakly supervised loss in experiments, making the whole system end-to-end complete.

The following is a summary of contributions:

(1) This paper introduces a type conversion matrix for node embedding in HINs, which can map nodes in heterogeneous networks to the same space for processing, facilitating the learnability of the network and ensuring the accuracy of node information. This solves the node embedding problem with multi-dimensional feature spaces in heterogeneous networks.
(2) This paper proposes a framework for attribute complementation of Heterogeneous Graph Neural Networks based on the Residual Attention mechanism (RA-HGNN), which can effectively solve the problems of missing attributes, over-smoothing and vanishing gradients of nodes in heterogeneous networks, and in a learnable manner solve the defect of artificially added attributes and can be easily combined with heterogeneous information networks.
(3) This paper conducted extensive classification and clustering experiments by combining the proposed framework with other heterogeneous models and evaluating the model's effectiveness. The datasets used in the experiments are DBLP, IMDB, and ACM, and the results show that models incorporated into this framework exhibit powerful performance and outperform other baselines.

2. Related work

2.1. Heterogeneous graph neural network

In the current world, a vast quantity of data may be represented using graph structures, and graph neural networks are advocated primarily for solving issues of this kind. One is that the spatial domain Hamilton, Ying, and Leskovec (2017), Jia et al. (2021), Liu et al. (2017), Wang, Xu, Chen, and Lin (2021) tries to apply the convolution operation directly to the graph, while the other is that the spectral domain Cai, Wang, and Wang (2021), Cao et al. (2020), Defferrard, Bresson, and Vandergheynst (2016) uses the eigenvalues and eigenvectors of the graph's laplacian matrix to investigate its attributes. For the spatial-GNN model, in DCNN (Atwood & Towsley, 2016), each node (or edge, or graph) is represented by H hop matrices, and each hop represents the adjacent information of the adjacent range. Therefore, the effect of getting local information is better, and the nodes that are gotten have a strong ability to represent. Also, for example, in GAT (Veličković et al., 2018) and GraphSAGE (Hamilton et al., 2017), the core concept of GAT (Veličković et al., 2018) is to implicitly apply different weights to various sorts of nodes, as opposed to manually defining the metrics between nodes as in MoNet (Xiao, Feng, Lin, Liu, & Zhang, 2018), and can be learned and used for semi-supervised learning. The network model that is more relevant to real-world problems today is the heterogeneous graph, so many have found so far that scaling from a homogeneous graph to a heterogeneous graph in the GNN model enables superior performance. For example HAE$_{GNN}$ (Li et al., 2021) addresses the problem of learning representation in heterogeneous networks by aggregating higher-order meta-path attribute information. Various attention mechanisms are utilized for different aggregate types of node neighbors in IHGAT (Liu et al., 2021), such as the attention mechanism for aggregating intent neighbors and the multi-head attention mechanism for aggregating transaction neighbors. However, there are missing attributes in the existing heterogeneous mapping, which leads to the missing of important node information. Therefore, HGNN-AC (Jin, Huo, Liang, & Yang, 2021) proposes an end-to-end heterogeneous network attribute complementation framework. AC-HEN (Wang, Yu, Huang, Zhao, & Dong, 2022) utilizes feature aggregation, structural aggregation and multi-view embedding fusion to achieve node attribute completion. HetReGAT-FC (Li, Yan, Fu, Zhao, & Zeng, 2023) utilizes type mapping to achieve attribute completion through residual GAT network for the heterogeneous nodes with different node types. AutoAC (Zhu et al., 2023) proposes differentiable attribute completion framework, which optimizes the heterogeneous model training and complementation process. In addition to this, HGCA (He et al., 2022) is an unsupervised algorithm that utilizes a comparative learning approach to achieve attribute complementation and representation learning for heterogeneous graphs.

Heterogeneous network attribute complementation frameworks face multiple challenges such as data sparsity, cross-type information fusion, attribute correlation, etc., which need to be considered comprehensively to design efficient and accurate models. However, the existing models face the problem that it is difficult to improve the model accuracy while reducing the model complexity, and it is often only possible to do one or the other. In this paper, an efficient attribute processing pre-training approach is adopted, and deep residual attention networks are utilized to achieve attribute complementation, which effectively compensates for the defects exposed above.

2.2. Graph embedding

Graph embedding reduces the dimensionality of the actual graph network from high to low dimensions. It represents the nodes of the graph as a low-dimensional vector space while keeping the network topology and node information. Initially, graph embedding techniques were mainly used for homogeneous graphs, including DeepWalk (Perozzi, Al-Rfou, & Skiena, 2014), node2vec (Grover & Leskovec, 2016), LINE (Tang et al., 2015), GCN (Kipf & Welling, 2016), and GraphSAGE (Hamilton et al., 2017). All of these are ways to learn how to represent a graph based on how similar two vertices are, and they only keep the features of a small part of the graph. Due to the emergence of heterogeneous graph networks, there has been corresponding research work on algorithms for heterogeneous graph embedding, but heterogeneous graphs predominate in practical business scenarios. The primary obstacles are: considering the information of various types of nodes and edges in heterogeneous networks; aggregation of neighbor nodes; encoding of heterogeneous content; and capturing deeper semantic information. For example, in metapath2vec (Dong, Chawla, & Swami, 2017), the way of embedding is similar to DeepWalk (Perozzi et al., 2014); a random walk of meta-path is used to obtain the sequence information of the nodes in a graph, and then a skip-gram (Mikolov, Chen, Corrado, & Dean, 2013) is used to learn the embedding representation of the nodes. There is a flaw in metapath2vec in that the node type is not considered; metapath2vec++ (Dong et al., 2017) is an upgrade of metapath2vec that primarily samples the negative samples of the core node's type. All of the approaches above for heterogeneous graph embedding are based on meta-path. At the same time, metagraph2vec (Zhang, Yin, Zhu, & Zhang, 2018) is a random walk directed by a meta-graph to build heterogeneous node sequences and then uses skip-gram technology to learn node embedding. The advantage of the heterogeneous graph embedding model described above is that it works well in parallel, but the disadvantage is also clear: it takes a lot of computation and memory to handle the low-dimensional embedding of each node. R-GCN (Schlichtkrull et al., 2018) projects node embeddings into several relational spaces using multiple weight matrices, which is different from metapath2vec and metagraph2vec and can capture the heterogeneity of the graph.

All the graph embedding methods mentioned above can have a good effect on the representation of node information and also show good performance indicators for processing downstream tasks. However, the problem of missing node attributes has not been solved. In the process of node embedding, the phenomenon of node information loss can occur, so that the model cannot achieve the optimal effect. Taking into account that nodes have different feature spaces, this paper introduces a type conversion matrix. Its goal is to ensure that the topological network's node attributes have integrity after embedding.

3. Methodology

This paper proposes attribute complementation for heterogeneous graph neural networks based on the residual attention mechanism(RA-HGNN) framework. The main idea behind this framework is to complement nodes with attributes for nodes without attributes based on the residual attention mechanism with the topological information of the heterogeneous graph as a guide.

3.1. Preliminary

This paper gives some terms related to heterogeneous graphs with their corresponding explanations, which will be used in some of the relevant mathematical expressions in this paper. As shown in Table 1.

Heterogeneous graph. $\mathcal{G}$ is the definition of a heterogeneous graph, expressed as $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{Y},\mathcal{S})$, $\mathcal{Y}$ specifies the kind of node set, $\mathcal{S}$ specifies the kind of edge set, each node $\boldsymbol{i} \in \mathcal{V}$ corresponds to a node type mapping function δ, $\delta : \mathcal{V} \rightarrow \mathcal{Y}$, and each edge $\boldsymbol{e} \in \mathcal{E}$ corresponds to a node type mapping function μ, $\mu : \mathcal{E} \rightarrow \mathcal{S}$.

Incomplete attributes. The completeness of node attributes inside a heterogeneous graph $\mathcal{G}$ refers to $\exists \mathcal{Y}' \subset \mathcal{Y}$ and $\mathcal{Y}' \neq \emptyset$, each node $\boldsymbol{i} \in \mathcal{V}$ corresponds to a node type mapping function φ, $\varphi : \mathcal{V} \rightarrow \mathcal{Y}'$ and $\mathcal{Y}'$ lacks attributes.

Table 1
Notations & Explanations.

Notations	Explanations
$\mathcal{G}$	A heterogeneous graph
$\mathcal{V}^+$	The set of nodes with attributes
$\mathcal{V}^-$	The set of nodes non-attributes
v	A node $v \in \mathcal{V}$
N_v^+	The set of node's neighbors in $v \in \mathcal{V}^+$
ϕ	A meta-path
h	Initial node attribute
h'	Node attribute after mapping
$\mathcal{M}_\phi$	Special types of transformation matrices
$\mathcal{A}$	Topological structure
X	The attribute of node
X^c	Attribute of nodes after attribute completion
H	The embedding of nodes

Residual attention mechanism. The primary task is performing residual connections on the I^{th} nodes and edges and then aggregating them. After embedding the graph topology has been completed, this approach may successfully prevent over-smoothing and improve the model's expressiveness via multi-head attention.

3.2. Overview

In this framework, the problem of some nodes' non-attributes in heterogeneous graphs is dealt with by using nodes with attributes to complete the attributes of non-attributed nodes through the topology of the heterogeneous graph. The Fig. 2 shows the entire framework structure. First, map different types of nodes to the same attribute type in a heterogeneous graph. The node embedding H is calculated through the corresponding topology $\mathcal{A}$, and then through the attention layer, which adds an edge residual connection operation to prevent over-smoothing and vanishing gradient, and calculates the sortable attention score between nodes, which is used to identify which nodes can add attributes to directly connected non-attribute nodes. Then, RA-HGNN does a weighted aggregation of the best-obtained attribute nodes $\mathcal{V}^+$ to complete the attributes of the nodes in $\mathcal{V}^-$. At the same time, some of the attributes with attribute nodes will be dropped in the framework, and then these attributes will be reconstructed through the RA-HGNN framework. The complete loss will be calculated between the dropped attributes and reconstructed attributes to verify the completed attributes' correctness. Finally, the node topology with the completed attributes will be fed to the HINs model. The final loss will be made up of the predicted loss and the completion loss, which will optimize the whole model in an end-to-end manner.

3.3. Topological embedding

Topological embedding may retain the topology structure, node semantic information, and attribute information in a heterogeneous network. However, the degree of similarity between adjacent nodes in a homogeneous graph is generally more significant than that between different types of nodes in a heterogeneous graph. Due to the nature of homogeneous graphs, the topology of the nodes and their attributes do not differ much. For example, the director is closely related to the actors used and the film created in the film and television fields. Since these nodes exist in the same field, the topological structure between the director, the actor, and the film will be similar and have the same attributes. For different types of nodes, there will be different topological structures and the problem of missing attributes. Through topology embedding, the attribute relationships between nodes can be better reflected. For example, the metapath2vec embedding method is based on the embedding method of a single meta-path, which ignores much important information. And embedding based on a meta-graph requires more memory space, and the execution efficiency is relatively poor. To get a better embedding structure, RA-HGNN maps all nodes into the same feature space, uses a random walk algorithm to get more detailed sequence information for multiple meta-paths, and then puts the obtained sequence information into the skip-gram model to learn node embeddings.

3.4. Attribute completion of residual attention mechanism

In earlier research, both GTN and MAGNN (Fu et al., 2020) addressed the problem of missing attributes in heterogeneous networks by aggregating information from neighboring nodes. Due to the different node types, most of the more similar nodes have a higher degree of correlation, and there are the same factors in semantic information, topology information, and attribute information, but this processing method does not guarantee that all adjacent nodes have similar information. Assuming that a node has a large number of neighbors, the importance of each neighbor's information to this node will decrease. In the RA-HGNN framework, the importance of node neighbors to the target node is actively learned, and the attribute information between nodes is aggregated through the residual attention mechanism. Then the attributes are further complemented by directly connected neighboring nodes with attributes for nodes without attributes.

Considering that there are a large number of different types of nodes in the heterogeneous graph, which also leads to different types of feature spaces, this framework needs to convert the different types of nodes into the same feature space within the type-specific conversion matrix involved here. This mapping operation is just a conversion for the different types of nodes in the heterogeneous graph. The specific mapping operation is as follows:

$$h'_i = M_i \cdot h_i \tag{1}$$

Here, h_i represents the node's initial attribute, and h'_i represents the mapped attribute. Various kinds of nodes are mapped to a particular feature space through the type-specific conversion matrix M_i.

At the framework's attention level, the significance of neighboring nodes is primarily calculated. For instance, if there exists a directly connected node pair (i, j), the importance value e_{ij}^{ϕ} of node j to node i may be estimated using the following method:

$$e_{ij}^{\phi} = att_{node}\left(h'_i, h'_j\right) \tag{2}$$

Here, ϕ represents a meta-path in the topology, h_i and h_j represent the topological embeddings of nodes i and j, where $j \in \mathcal{V}^+$, $att_{node}(\cdot)$ represents the execution of the attention mechanism and is available for all node pairs.

In this framework, since the contribution of the first-order neighbor j to i is the largest among all the neighbors of node i, for e_{ij}^{ϕ} calculated above, node j represents the attribute mapping of the first-degree neighbors of the node i, where $i \in \mathcal{V}^+$ and $j \in N_v^+$. Consequently, this measure can filter a large number of irrelevant attribute nodes, and it minimizes the calculation of nodes in $\mathcal{V}^-$ and improves a portion of the efficiency by using the masked attention mechanism:

$$e_{ij}^{\phi} = \sigma\left[\left(h'_i\right)^T W h'_j\right] \tag{3}$$

W represents a parameter matrix, and σ is the activation function.

The above calculation can obtain the attention scores of all nodes with attributes and their direct neighbors. Then the softmax activation function is used to determine the weight normalization parameters:

$$\hat{a}_{ij} = \mathrm{softmax}\left(e_{ij}^{\phi}\right) = \frac{\exp\left(e_{ij}^{\phi}\right)}{\sum_{j \in N_i^+} \exp\left(e_{ij}\right)} \tag{4}$$

Then, through the obtained weight parameter $\hat{a}_{ij}$, RA-HGNN will weigh the aggregated attribute information for node i based on the final attention score a_{ij}:

$$x_i^C = \sum_{j \in N_v^+} a_{ij} x'_j \tag{5}$$

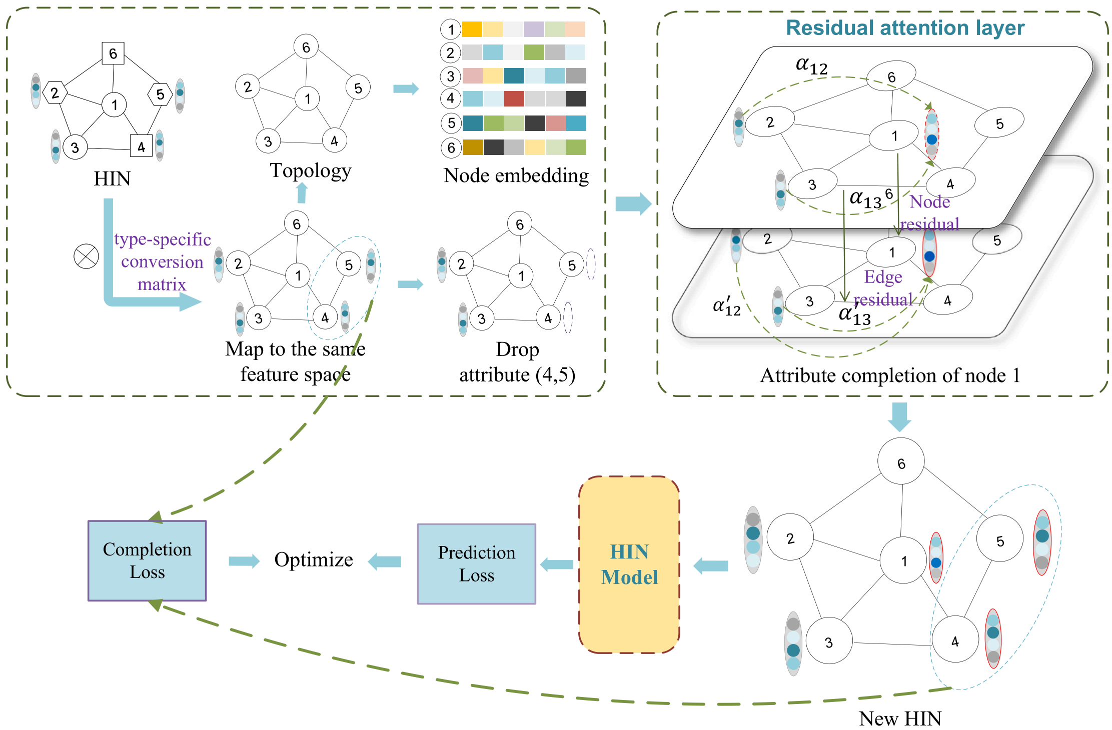


Fig. 2. A description of the framework. Given a heterogeneous graph with different types of nodes, first map to the same feature space, ensure that the nodes have the same attribute dimensions, and compute the node embeddings via the network topology A; then, randomly drop some attributes and perform attribute complementation. The essence of attribute complementation is to aggregate the weighted aggregation of neighbor nodes with attributes by residual attention, and the weights are determined by the attention scores. After attribute completion, and the heterogeneous graph topology with attributes is input into the HINs model, the model is optimized using predictive loss and completion loss back propagation, and the whole model is completed end-to-end.

Finally, this paper uses multi-head attention to boost a model's expression capacity and uses this method to aggregate the attributes of node i:

$$x_i^C = \text{mean}\left(\sum_{k}^{K} \sum_{j\in N_i^+} a_{ij} x'_j\right) \tag{6}$$

K indicates that this paper conducted K distinct attention processes, and mean$(\cdot)$ refers to that averaged these results.

In heterogeneous graphs, there will be some exceptional cases. For example, node i does not have any neighbors with attributes. That is, the neighbors of a node i do not have attributes (i.e. $N_i^+ = \varnothing$), In practice, these circumstances are mostly nonexistent. Hence, its effect on results is minimal.

In some GNNs (Ding et al., 2021; Kipf & Welling, 2016), there are always some over-smoothing and vanishing gradient problems, and this paper uses a residual connection to deal with such problems.

Node residual. Since the node attribute dimensions are the same, performing pre-activation residual connection on the input node attribute and aggregating the node attributes, the specific implementation is as follows:

$$h_i'^{(m+1)} = \sigma\left[\sum_{j\in N_i^+} a_{ij}^{(m+1)} W^{(m+1)} \left(h'_j\right)^m + \left(h'_i\right)^m\right] \tag{7}$$

The above case is designed with a pre-trained residual connection that aggregates over the $(m+1)^{th}$ layer, considering when the node layer changes.

When the dimension changes, an additional learnable linear transformation matrix $W_{\text{res}}^{(m+1)}$ is used here.

$$h_i'^{(m+1)} = \sigma\left[\sum_{j\in N_i^+} a_{ij}^{(m+1)} W^{(m+1)} \left(h'_j\right)^m + W_{res}^{(m+1)} \left(h'_i\right)^m\right] \tag{8}$$

Edge residual. The raw attention scores $\hat{\alpha}_{ij}$ between nodes are obtained by Eq. (4), and for the different layers, this paper adds the residual connections:

$$a_{ij}^{(m+1)} = (1-\beta)\hat{\alpha}_{ij}^{(m+1)} + \beta a_{ij}^{(m)} \tag{9}$$

where hyperparameter $\beta \in [0,1]$ is a scaling factor.

In particular, this paper performs K independent attention mechanisms and aggregates these results as the final expression:

$$a_{ijk}^{(m+1)} = (1-\beta)\hat{\alpha}_{ijk}^{(m+1)} + \beta a_{ijk}^{(m)} \tag{10}$$

$$\hat{h}_{ik}^{(m+1)} = \sum_{j\in N_i^+} a_{ijk}^{(m+1)} W_k^{(m+1)} \left(h'_j\right)^m \tag{11}$$

$$\boldsymbol{h}_i'^{(m+1)} = \sigma\left(\|_{k=1}^{K} \hat{\boldsymbol{h}}_{ik}^{(m+1)} + \boldsymbol{W}_{\text{res}(k)}^{(m+1)} (\boldsymbol{h}'_i)^m\right) \tag{12}$$

$$\boldsymbol{h}_i^{(M)} = \frac{1}{K}\sum_{k=1}^{K} \hat{\boldsymbol{h}}_{ik}^{(M)} \tag{13}$$

Here, $\|$ represents the concatenation operation, and $a_{ijk}^{(m+1)}$ is the attention score calculated by the k^{th} linear transformation $W_k^{(m+1)}$. Since the output dimension cannot be divided precisely by K, an averaging method is used to determine the final number of layers (M^{th}).

In summary, in traditional deep networks, the gradient shrinks during back propagation, resulting in extremely small parameter updates for shallower layers, which makes it difficult to train effectively. By summing the input attributes with the output attributes through node residuals and edge residuals (Eqs. (7)–(13)), the path of the original information is preserved, mitigating the attenuation of the gradient as it propagates. This means that during back propagation, even if the gradient decreases in some layers, it is still possible to pass the gradient directly from shallower layers through the residual connection, avoiding the gradient vanishing problem. Residual connections allow the network to gradually adjust the original inputs by learning the residuals, thus making it easier to learn the constant mapping (i.e., inputs and outputs are equal) and the residual part. This allows the network to selectively learn hard-to-capture changes and learn simpler changes as residuals. This helps the network to fit the data better and thus improves the expressive power of the network.

The Algorithm 1 implements the attribute completion part of the residual attention mechanism in the framework.

Algorithm 1 RA-HGNN

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
```
for each V_i ∈ {V^-, V^+} do
    Map different nodes to the same feature space by Eq. (1);
    Topology A computes node embedding information;
    for each h'_i do
        The weight normalization parameters are obtained a_ij by Eq. (4);
        The (m+1)^th layer attention score is obtained via the edge residual Eq. (9);
        Node feature of the (m+1)^th layer are obtained by node residual Eq. (8);
        Eq. (13) obtains the final feature expression;
        Get the aggregation of attributes of the node by Eq. (6);
    end for
end for
```
Output: The node attribute information x_i^C after residual connection is calculated;
```
The prediction loss L is calculated;
Back propagation is used to optimize the model;
```
Output: All the knowns.

3.5. Dropping some attributes

The model framework this paper proposes is based on a residual attention mechanism to complete the attributes of nodes in heterogeneous graphs. What this paper wants to aim for is to enhance the model's performance. When nodes are completed, all nodes with first-order attributes are used. How to judge the effect of this learnable result?

In response to the above questions, in RA-HGNN, a part of the nodes in $\mathcal{V}^+$ is randomly selected, the attributes are dropped, and then the RA-HGNN framework is used to complete the attributes in $\mathcal{V}^-$ again. At the same time, attribute completion is performed on nodes that drop attributes. The completion loss calculated by this whole process is derived from the drop attribute and the reconstruction attribute.

Through the above analysis, the node in $\mathcal{V}^+$, will be divided into two parts $\mathcal{V}^+_{drop}$ and $\mathcal{V}^+_{re}$, for the node in $\mathcal{V}^+_{drop}$, a parameter α will be set as the drop rate, expressed as $|\mathcal{V}^+_{drop}| = \alpha|\mathcal{V}^+|$, for the nodes with attributes in the heterogeneous graph, the set of nodes with dropped attributes is $\mathcal{V}^+_{drop}$, which is reconstructed by the set of nodes in $\mathcal{V}^+_{re}$. The reconstructed attribute node i is represented as:

$$x_i^C = \text{mean}\left(\sum_k^K \sum_{j\in\mathcal{V}^+_{re}\cap\mathcal{V}^+_n} a_{ij} x'_j\right) \tag{14}$$

Masked attention is also used here, where K and the function $\text{mean}(\cdot)$ are the same as mentioned above.

To reconstruct the node attributes in the set $\mathcal{V}^+_{drop}$, the goal of this paper is to be as close to the original attribute value as possible, so optimization scheme of this paper is to use weak supervision loss. The Euclidean distance of attribute nodes to express completion loss:

$$\mathcal{L}_{\text{completion}} = \frac{1}{|\mathcal{V}^+_{drop}|} \sum_{n\in V^+_{drop}} \sqrt{\left(X_n^C - X_n\right)^2} \tag{15}$$

3.6. Analysis of the HIN model

Through RA-HGNN, the attribute completion of the nodes in the sets $\mathcal{V}^+_{drop}$ and $\mathcal{V}^-$ is completed. The completed nodes are defined as:

$$X^{new} = \left\{X_n^C, X_m^C, X_r \mid \forall n \in \mathcal{V}^-, \forall m \in \mathcal{V}^+_{drop}, \forall r \in \mathcal{V}^+_{re}\right\} \tag{16}$$

The designed completion framework is mainly applied to different HINs models, and during the execution of this framework, it will not affect the topology of the graph, so this paper will get the node X^{new} and the topology of the graph structures can be sent directly to the HINs model:

$$\tilde{Y} = \Delta\left(A, X^{new}\right) \tag{17}$$

$$\mathcal{L}_{\text{prediction}} = f(\tilde{Y}, Y) \tag{18}$$

$$\mathcal{L} = \varepsilon\mathcal{L}_{completion} + \mathcal{L}_{prediction} \tag{19}$$

Here, Δ represents different HINs models, A represents the topology, Y represents the label, f represents the loss function, and ε represents a weight parameter that balances the two parts.

Finally, this paper applies the framework to different HINs models, and the final loss function is expressed as the prediction loss for labels and the complete loss for attributes. Excellent performance can be obtained through the combination of this framework and the model, and the performance is excellent in the backpropagation process.

4. Experiments

4.1. Datasets

To more clearly describe the information in the heterogeneous graph, this paper used three different datasets for the representation: a subset of DBLP, ACM, and IMDB. The analysis is described in Table 2.

(1) DBLP. This paper selected a portion of the node information in the DBLP dataset, including 14,458 papers (P), 4057 authors (A), 20 venues (V), and 8694 terms (T). The authors' attribute is the bag-of-words representation of keywords, the venues' attribute is a one-hot vector, and the terms' attribute has no pre-trained word vectors (Pennington, Socher, & Manning, 2014). Only the papers' attributes are directly from the dataset.

(2) ACM. This paper selected a portion of the node information in the ACM dataset, including 4108 papers (P), 7235 authors (A) and 59 subjects (S). The attribute of paper is the bag-of-words representation of keywords, and the attribute vectors of authors and subjects are derived from the attribute vectors of directly related papers. Only the papers' attributes come directly from this dataset in this dataset.

(3) IMDB. This paper selected a portion of the node information in the IMDB dataset, including 4801 movies (M), 5796 actors (A) and 2304 directors (D). There are three categories for movies: action, comedy, and drama. In this dataset, the movies' attributes are represented by a bag-of-words, the actors' and directors' attribute vectors are obtained directly from the movies' attributes vector, and only the movies' attributes come directly from the dataset.

4.2. Baselines

In the experiment, different models will be used to compare with the model combined with this framework. This framework is connected with the current state-of-the-art model MAGNN, defined as RA-MAGNN. Then RA-MAGNN is combined with the original MAGNN, which conducts comparative experiments and processes the data. The specific baselines are as follows.

Table 2
Datasets.

Dataset	Node	Edge	Attribute
ACM[a]	paper(P):4,019 author(A):7,167 subject(S):60	paper-paper:9,615 paper-author:13,407 paper-subject:4,019	paper:raw author:handcrafted subject:handcrafted
DBLP[b]	author(A):4,057 paper(P):14,328 term(T):7,723 venue(V):20	author-paper:19,645 paper-term:85,810 paper-venue:14,328	author:handcrafted paper:raw term:handcrafted venue:handcrafted
IMDB[c]	movie(M):4,278 director(D):2,081 actor(A):5,257	movie-director:4,278 movie-actor:12,828	movie:raw director:handcrafted actor:handcrafted

[a] http://dl.acm.org/.
[b] https://dblp.uni-trier.de/.
[c] https://www.imdb.com/.

(1) Metapath2vec (Dong et al., 2017): A meta-path based random walk is used to construct the heterogeneous neighborhood of each vertex, and then a skip-gram is used to complete the model of vertex embedding. This paper will test all the meta-paths of the model and record the best experimental results.
(2) GCN (Kipf & Welling, 2016): A semi-supervised graph convolutional network is a homogeneous GNN that can work directly on graphs and exploit the structural information of graphs. This paper tests the effect of GCN based on different meta-paths in homogeneous graphs and records the best results.
(3) HAN (Wang et al., 2019): It is a heterogeneous GNN. This model is divided into two layers in the attention mechanism: node layer attention and semantic layer attention, and node embedding is performed through the attributes of neighbor-based meta-path aggregation.
(4) MAGNN (Fu et al., 2020): It is a heterogeneous GNN that solves the embedding learning problem of heterogeneous graphs. Model components include node information translation, intra-meta-path and inter-meta-path aggregation.
(5) GAT (Veličković et al., 2018): It is a homogeneous GNN. It does not depend on the graph's structure and uses masked attention to complete the model operation. This paper trains GAT through a meta-path and records the results.
(6) MAGNN-AC (Jin et al., 2021): It is MAGNN that combines the HGNN-AC (Jin et al., 2021) framework for heterogeneous GNNs to solve the problem of missing attributes in heterogeneous networks.
(7) AC-HEN (Wang et al., 2022): It is a generalized attribute complementation framework for heterogeneous networks that utilizes feature aggregation, structure aggregation, and multi-view embedding fusion to achieve attribute complementation.
(8) HetReGAT-FC (Li et al., 2023): It is a deep heterogeneous network that complements the feature information in the heterogeneous graph.

4.3. Implementation details

The baseline models described in the preceding section maintain their initial configuration settings and do not change. For this framework (RA-HGNN), set the parameters' learning rate to 0.005; this paper's loss rate when using mask attention is 0.5. In the extended multi-head attention mechanism, the number of attention heads is set to K=8. The attribute drop rate for nodes with attributes is set to 0.3. For a fair comparison, this paper sets this embedding dimension to 64 for all models. The node parameters of the dataset are consistent with (Fu et al., 2020; Jin et al., 2021; Wang et al., 2019), and the experimental data in this paper will be referred to in Tables 3 and 4.

4.4. Classification

In the graph, the classification of nodes is a very critical task. In this framework, node classification is usually the task of classifying nodes with attributes and without attributes, so this paper needs to set up two different datasets to operate this task. Among them, the nodes of the first classification are nodes with attributes, and here in this paper, the datasets for experiments on ACM and IMDB are chosen. Another classified node is the node without original attributes, and this paper uses the DBLP dataset for experiments. The three datasets are processed using the baseline model, and the effectiveness of the framework proposed in this paper is demonstrated by combining it with the MAGNN model and simultaneously comparing it with the currently optimal attribute complementation networks (HGNN-AC,AC-HEN and HetReGAT-FC). In order to improve the credibility of the classification task, we chose a linear support vector machine (SVM) (Suykens, 2001) and set the training ratio from 1% to 80%. Since in the semi-supervised experiment, the labels in the training set and the test set are involved in the training of the model, in order to maintain consistency, only the labels in the test set are put into the linear SVM. Due to the significant data differences for different graph structures, this paper repeated the test 5 times and recorded the average value of Macro-F1 and Micro-F1.

Table 3 shows the results of experiments after performing the node classification task in the datasets ACM, DBLP, and IMDB. Paper nodes and movie nodes are nodes with original attributes in the ACM and IMDB datasets, respectively. In contrast, the other types of nodes are nodes without original attributes, where paper and movie are the nodes performing the classification task. By combining to maintain consistency, it can be seen through the results that there is a significant reduction in the error rate compared to other models and that RA-MAGNN works better overall. Different training ratios are adopted in the DBLP dataset, and the author node without the original attribute is used to perform classification task. RA-MAGNN completes the attributes of author, venue, and term nodes through the paper node. Overall, as shown in Tables 3, the baseline MAGNN-AC and MAGNNN are superior to the other baseline models, and more importantly, RA-MAGNN improves accuracy by 0.04%–4.66% compared to MAGNN-AC and 0.03%–3.35% compared to MAGNN for the same training ratio. Meanwhile, in Table 3, the performance of RA-HGNN is also significantly improved compared with the complementary models AC-HEN and HetReGAT-FC. It can be seen that RA-HGNN has a significant advantage in representing node attributes and shows strong stability when combined with the HINs model.

Through the above analysis, according to the topological relationship, this paper performs attribute completion on nodes through a residual attention mechanism, which shows superior performance. For a particular node, this approach can learn the attention score of this node's neighbors to this node efficiently instead of setting an importance value manually, which will ignore the importance of this node's neighbors. Therefore, this framework can address the problems of missing node attributes and over-smoothing in heterogeneous graphs.

In order to verify that this framework can be widely applied with other heterogeneous network models, therefore, HAN and MAGNN networks are selected as the baseline heterogeneous models and are combined with them using the attribute complementation frameworks AC-HEN, HGNN-AC, HetReGAT-FC, and RA-HGNN respectively with the training ratio set to 40%, and verifying their respective performances. The performance of each is verified. As shown in Table 5, HAN_{HEN}, $HAN_{HGNN-AC}$, HAN_{FC}, and HAN_{RA} are AC-HEN, HGNN-AC, HetReGAT-FC, and RA-HGNN variant models, respectively, and it can be seen that HAN_{RA} has the largest performance improvement in the three different datasets, when compared with the optimal performance of HAN_{FC} in the baseline. datasets, with the biggest performance improvement of 1.57% and 1.11% for Macro-F1 and Micro-F1, respectively. Compared with the optimal performance

Table 3
Results (%) on ACM, IMDB and DBLP datasets on the node classification task.

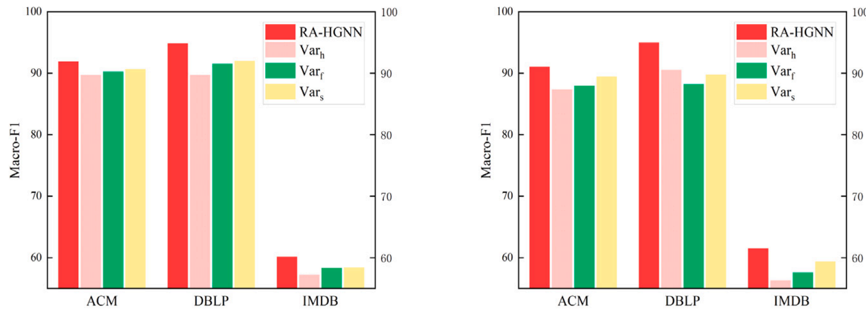
Datasets	Metrics	Training	Without attribute completion					With attribute completion			
			Metapath2vec	GCN	HAN	GAT	MAGNN	MAGNN-AC	AC-HEN	HetReGAT-FC	RA-MAGNN
ACM	Macro-F1	1%	35.23	66.83	86.95	88.58	85.58	84.27	85.11	85.37	**88.93**
		5%	42.37	72.45	88.65	89.20	86.12	85.47	86.72	86.51	**89.25**
		10%	44.29	70.79	89.39	89.29	86.60	86.23	86.95	87.69	**89.51**
		20%	69.95	70.41	89.40	**89.59**	88.01	88.44	88.15	88.25	89.55
		40%	71.15	70.82	89.79	89.77	89.42	89.93	89.33	89.51	**90.69**
		60%	71.47	69.67	89.51	89.72	90.39	90.67	90.66	90.42	**91.52**
		80%	72.18	67.23	90.63	89.42	90.79	91.08	90.82	90.93	**91.96**
	Micro-F1	1%	63.18	71.74	87.37	88.60	85.95	84.82	85.05	85.33	**88.96**
		5%	68.88	74.95	88.62	89.10	86.24	85.52	86.73	87.21	**89.26**
		10%	70.29	71.40	89.32	89.19	86.67	86.19	87.52	87.39	**89.34**
		20%	72.12	74.02	89.22	**89.47**	88.08	88.42	88.25	88.26	89.41
		40%	73.17	74.57	89.64	89.65	89.48	89.95	89.51	89.63	**90.70**
		60%	73.65	74.10	89.33	89.60	90.42	90.66	90.46	90.77	**91.55**
		80%	74.14	72.69	90.54	89.29	90.80	91.05	90.83	90.92	**91.98**
IMDB	Macro-F1	1%	35.23	39.72	52.49	49.52	50.78	49.75	49.88	50.57	**52.63**
		5%	42.37	42.95	56.16	53.08	54.28	53.60	54.49	54.41	**56.29**
		10%	44.29	43.70	57.02	53.61	56.39	55.25	55.35	56.81	**57.23**
		20%	46.42	44.75	50.00	54.81	58.11	58.17	58.33	58.19	**58.26**
		40%	47.70	45.26	52.71	55.09	59.39	59.26	59.41	59.32	**59.46**
		60%	48.25	46.72	54.24	55.71	**59.97**	59.45	59.82	59.91	59.95
		80%	48.73	47.13	54.38	55.40	60.02	60.08	60.08	60.11	**60.19**
	Micro-F1	1%	39.55	44.01	54.38	51.32	51.62	50.47	51.36	51.87	**54.69**
		5%	44.33	46.41	56.74	53.73	54.46	53.77	54.22	55.35	**56.90**
		10%	46.15	47.02	57.35	54.14	56.53	55.48	55.67	55.41	**57.42**
		20%	48.08	47.44	55.73	55.02	58.16	57.27	58.14	58.17	**58.27**
		40%	49.55	47.62	57.97	55.29	59.46	59.18	59.33	59.31	**59.49**
		60%	50.06	48.49	58.32	55.91	60.05	59.58	59.57	**60.08**	59.98
		80%	50.68	48.73	58.51	55.67	60.15	60.13	60.11	60.05	**60.24**
DBLP	Macro-F1	1%	88.76	86.99	89.37	32.68	92.45	92.69	92.16	92.38	**92.73**
		5%	90.49	89.03	90.83	57.20	92.44	93.10	93.29	93.36	**93.70**
		10%	91.09	89.53	91.24	64.57	92.44	93.18	92.98	93.34	**93.97**
		20%	91.50	90.06	92.24	66.92	93.13	93.21	93.26	93.71	**94.20**
		40%	92.55	90.37	92.40	73.23	93.23	93.35	93.55	93.83	**94.35**
		60%	93.25	90.57	92.80	77.17	93.57	93.38	93.75	94.02	**94.38**
		80%	93.48	90.74	93.08	78.20	94.10	94.02	94.22	94.19	**94.63**
	Micro-F1	1%	89.91	87.55	90.12	48.74	93.11	93.25	92.54	93.21	**93.29**
		5%	91.19	89.58	91.49	70.79	93.02	93.51	93.22	93.01	**94.14**
		10%	91.74	90.02	91.88	75.90	93.02	93.55	92.37	94.02	**94.39**
		20%	92.14	90.53	93.11	76.98	93.61	93.60	94.14	94.33	**94.60**
		40%	93.09	90.83	93.30	79.61	93.68	93.37	93.84	94.01	**94.73**
		60%	93.76	91.01	93.70	81.62	93.99	93.75	94.06	94.53	**94.75**
		80%	93.94	91.15	93.99	82.22	94.47	94.17	94.68	94.52	**94.98**

Table 4
Quantitative results (%) on the node clustering task.

Datasets	Metrics	Metapath2vec	GCN	HAN	GAT	MAGNN	MAGNN-AC	AC-HEN	HetReGAT-FC	RA-MAGNN
ACM	NMI	21.22	51.40	61.56	57.29	62.51	64.93	64.82	64.97	**65.21**
	ARI	21.00	53.01	64.39	60.43	67.23	68.52	68.84	69.05	**69.88**
IMDB	NMI	1.20	5.45	10.87	8.45	12.87	14.29	14.22	14.07	**14.31**
	ARI	1.70	4.40	10.01	7.46	11.98	14.09	14.01	14.10	**14.12**
DBLP	NMI	74.30	75.01	79.12	71.50	77.64	79.49	79.51	79.35	**79.59**
	ARI	78.50	80.49	84.76	77.26	82.60	84.83	84.51	84.73	**84.92**

Table 5
The effect of different attribute completion on HGNNs (training ratio is 40%).

Datasets	ACM		IMDB		DBLP	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
HAN	89.79	89.64	52.71	57.97	92.40	93.30
HAN_{HEN}	90.11	90.27	57.14	58.26	92.57	93.69
$HAN_{HGNN-AC}$	89.96	89.73	57.09	58.03	92.28	93.48
HAN_{FC}	90.31	90.35	57.15	58.22	93.26	93.83
HAN_{RA}	**90.54**	**90.61**	**58.72**	**59.38**	**93.95**	**94.68**
$MAGNN$	89.42	89.48	59.39	59.46	93.23	93.68
$MAGNN_{HEN}$	90.43	59.52	59.01	59.43	93.52	93.77
$MAGNN_{HGNN-AC}$	89.93	89.95	59.26	59.18	93.35	93.37
$MAGNN_{FC}$	90.58	89.51	59.43	59.46	93.78	94.21
$MAGNN_{RA}$	**90.69**	**90.70**	**59.46**	**59.49**	**94.35**	**94.73**



(a) Macro-F1 on node classification (b) Micro-F1 on node classification

Fig. 3. Experimental study of ablation of RA-HGNN.

of $MAGNN_{RA}$ and $MAGNN_{FC}$ in the baseline, the maximum performance improvement of Macro-F1 and Micro-F1 is 0.57% and 1.19%, respectively. The above results fully demonstrate that the method proposed in this paper greatly improves the performance of the two HGNNs.

4.5. Ablation experiment

In order to validate each component of the RA-HGNN framework, this study will complete ablation experiments on the following variant models. In this experimental section, the underlying experimental parameter settings and model structure will remain the same, although the individual components are changed. The Macro-F1 and Micro-F1 performance of all variant models on the three datasets is shown in Fig. 3.

- Var_f: This variant model does not perform the same dimension mapping on the initial input node dimensions, which means that h_i and h_j are used as input attributes.
- Var_h: This varionverts the residual attention aggregation to the feature aggregation of traditional HGNN to verify the performance of residual attention.
- Var_s: This variant model eliminates the process of attribute complementation using multiple heads of attention and uses the self-attention mechanism instead.

Effect of feature mapping. The comparison between RA-HGNN and Var_f can show the importance of node dimension mapping in preprocessing. As can be seen from Fig. 3, the performance of RA-HGNN is significantly higher than that of Var_f in three different datasets, respectively, which suggests that the node feature dimensionality mapping transformations proposed in this study play a key role in model training.

Effect of residual attention. The comparison between RA-HGNN and Var_h can show the importance of residual attention on neighborhood feature aggregation. From Fig. 3(b), it can be seen that the performance of Var_h is lagging behind compared with RA-HGNN in the three datasets, and at the same time, it shows that the attribute aggregation module based on the mechanism of residual attention designed in this study has an obvious advantage in capturing neighbor attributes.

Effect of multi-head attention. the comparison between RA-HGNN and Var_s shows the importance of multi-head attention in capturing multi-neighborhood relationships. As shown in Fig. 3, again in all datasets, the RA-HGNN model is superior to the traditional self-attention mechanism.

4.6. Clustering

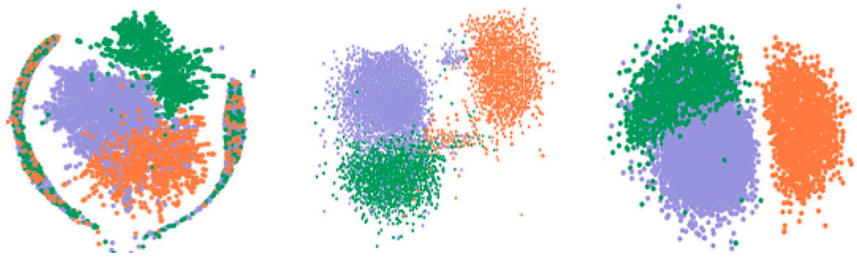
This paper uses the clustering task to compare different models and use the classical algorithm K-means to cluster the nodes obtained after training, where K is the number of classifications. And utilizing NMI and ARI to assess the clustering task outcomes, this paper performed the experiment 5 times and reported the averaged results in Table 4.

In order to more effectively describe the clustering effect of the baseline in different datasets, this paper adopts the NMI (Normalized Mutual Information) and ARI (Adjusted Rand Index) metrics that are applicable to the clustering problem of different categories of data. From Table 4, it is clear that the MAGNN-AC model outperforms other models not combined with the proposed framework on the ACM, DBLP and IMDB datasets. Then this paper combine the new framework with the MAGNN model, and the experimental findings may have obtained excellent outcomes. Therefore, different classification types can be distinguished in the clustering task. The advantage of RA-MAGNN over MAGNN is that the attributes of the nodes in the initial MAGNN are not obtained by learning the model itself but are added by handicraft, which will lead to the result of the clustering is not ideal. The framework this paper propose is to learn the attributes of missing nodes in a learnable way based on the topology structure and obtain the weighted information between each node through a residual attention mechanism to increase the accuracy of downstream tasks and the model's performance.

4.7. Visualization

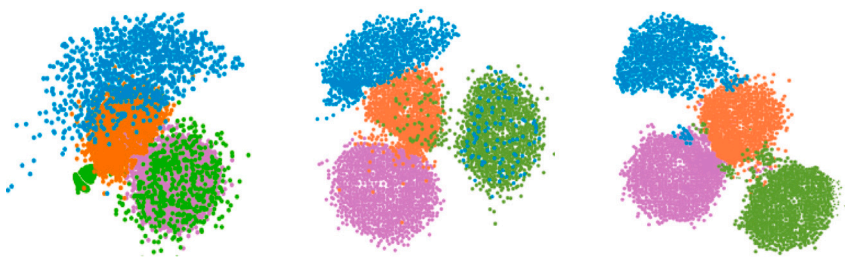
This paper presents a more intuitive comparison of the models by way of a visualization task. The MAGNN, MAGNN-AC, and RA-MAGNN models are embedded with nodes on the ACM, DBLP, and IMDB datasets, respectively, and visualized in two dimensions. Then, due to a large number of nodes in the dataset, paper nodes in the ACM dataset, paper nodes in the DBLP dataset, and movie nodes in the IMDB dataset were selected for embedding analysis in this paper and were color classified according to the nodes.

As shown in Figs. 4–6, in the three datasets, MAGNN works better than the original dataset, but does not perform as well as RA-MAGNN, where some of the different categories of papers and movies cross each other and the boundaries are blurred. The RA-MAGNN combined with this framework can show a clearer division boundary and can clearly distinguish different categories. The results show that accurate attribute information is beneficial for performing heterogeneous network work. MAGNN uses a manual process to get the attributes of nodes, which can lead to inaccurate attributes and bad performance. Using a multi-layered residual attention mechanism to complement the attributes in a learnable manner, guided by a topological structure, greatly improves the performance of the model and enables accurate classification of papers as well as films in different domains.



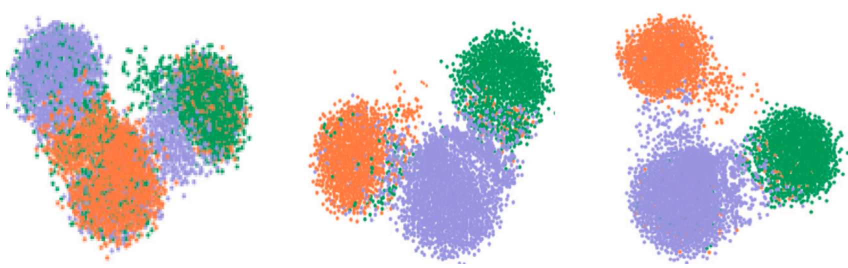
(a) Original graph (b) MAGNN-AC (c) RA-MAGNN

Fig. 4. Visualization of the paper nodes of embeddings in the ACM dataset.



(a) Original graph (b) MAGNN-AC (c) RA-MAGNN

Fig. 5. Visualization of the paper nodes of embeddings in the DBLP dataset.



(a) Original graph (b) MAGNN-AC (c) RA-MAGNN

Fig. 6. Visualization of the paper nodes of embeddings in the IMDB dataset.

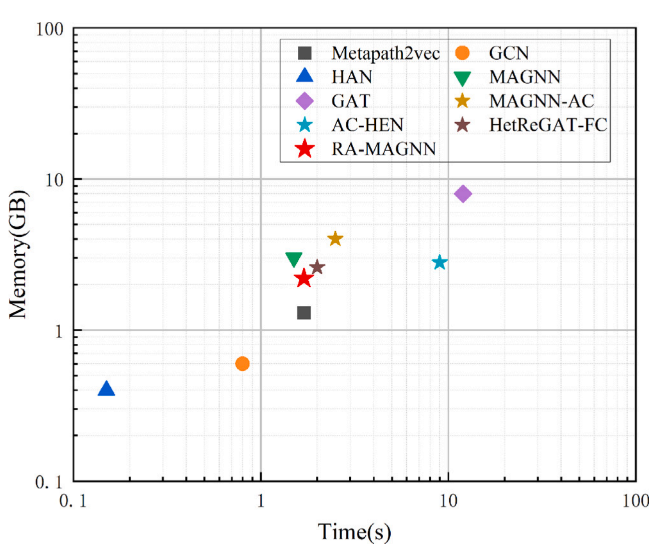
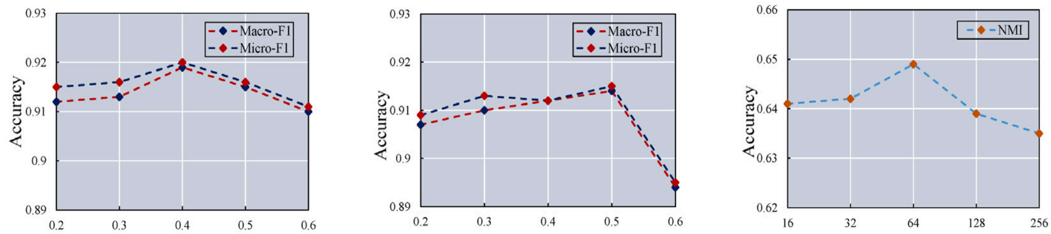


Fig. 7. Complexity comparison of RA-HGNN with other baseline models.

4.8. Complexity analysis

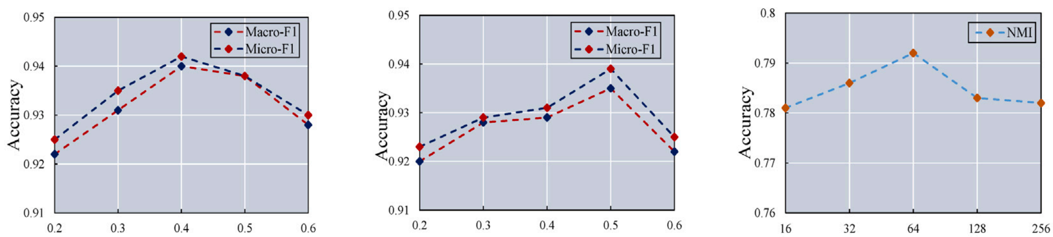
In order to verify the execution efficiency of this model, the execution time as well as the memory consumption of this model during the run of the ACM dataset are analyzed in this section, all the experiments are implemented on a server configured with Intel Core i9-12900K GPUs, and the specific results are shown in Fig. 7, which compares all baseline models involved in the experiments, with the x-axis denoting the average runtime for one iteration, and the y-axis denoting the memory consumed for one iteration of the model in which the iteration is performed.

One of them, RA-MAGNN, has a small improvement in runtime and memory consumption compared to MAGNN, which is clearly acceptable. However, RA-MAGNN is degraded compared to the attribute complementation class algorithms HGNN-AC, AC-HEN, and HerReGAT-FC. This study analyzes the possible reasons for this phenomenon: in comparison with HGNN-AC and AC-HEN, which also perform pre-training, it is seen that the execution time and memory consumption of the present model are lower, which once again proves the high efficiency of the mapping matrices; the HerReGAT-FC model needs to perform additional computations each time when mapping the node types, which in turn reduces the operational efficiency.



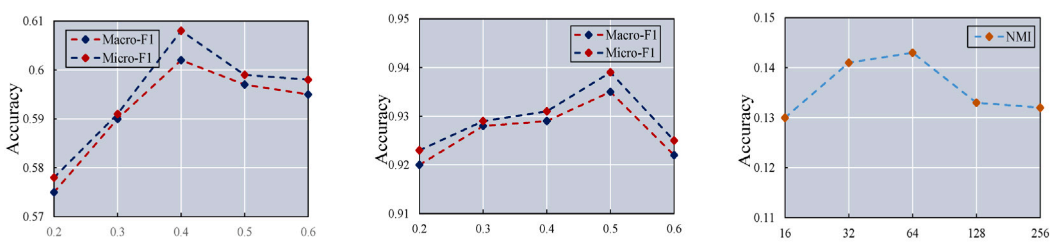
(a) Weighted coefficient λ (b) Divided ratio α (c) Embedding dimension Z

Fig. 8. Parameters analysis of RA-HGNN in the ACM dataset w.r.t. Weighted coefficient λ, Divided ratio α, Embedding dimension Z.



(a) Weighted coefficient λ (b) Divided ratio α (c) Embedding dimension Z

Fig. 9. Parameters analysis of RA-HGNN in the DBLP dataset w.r.t. Weighted coefficient λ, Divided ratio α, Embedding dimension Z.



(a) Weighted coefficient λ (b) Divided ratio α (c) Embedding dimension Z

Fig. 10. Parameters analysis of RA-HGNN in the IMDB dataset w.r.t. Weighted coefficient λ, Divided ratio α, Embedding dimension Z.

4.9. Parameters experiments

This subsection studies the effect of parameter changes in the ACM, DBLP and IMDB dataset on the results in detail and displays the Macro-F1 and Micro-F1 of node classification in the form of a line graph. According to the statistics of the division ratio of weight parameters and node attributes, the score corresponding to each variable is the average result value according to the training ratio. At the same time, the embedding dimension was adjusted and the experimental results were recorded. Please see Figs. 8–10 for details. Since the overall change trend of the experimental results is consistent across the three datasets, the following will be analyzed from the ACM dataset.

(1) Weight parameter λ: In this paper, different weight parameters are set to test and observe the impact on the performance of the model. The result obtained is Fig. 8(a). It is evident from the image that in the process of increasing the weight parameter value, the classification results show growth and then a downward trend. Therefore, it can be seen that too small and too high parameter settings will affect the model's performance. This paper needs to choose appropriate parameters to achieve the optimal effect.

(2) Divided ratio α: This paper tested for different α values. According to Fig. 8(b), it is evident that when the α value increases, the model's performance improves at first and then drops. For RA-HGNN, discarding too many or too few node attributes will produce unsatisfactory results, and the model needs a suitable α value to optimize the model's performance.

(3) Embedding dimension Z: This paper also tested the embedding dimension Z. The results are shown in Fig. 8(c). The findings show that when the embedding dimension grows, performance improves at first, then drops. The reason may be that there will be a significant deviation when the dimension is too low. When the dimension is too high, the embedding will have too large a variance. Therefore, choosing 64 embedding dimensions is the best performance for the model.

5. Conclusion and future work

This paper proposes the RA-HGNN framework to solve the common problem of embedding missing information and attributes missing in heterogeneous graphs through a learnable approach. Initially, the nodes in the heterogeneous graph are mapped into the same feature space for node topology embedding. Then the residual attention mechanism is used to add attributes to the non-attribute nodes by aggregating the weight fractions of the nodes with attributes according to the embedding results. Finally, through back-propagation optimization, an end-to-end model is obtained. This paper can solve the over-smoothing and vanishing gradient problems in models by combining the framework with the HINs model. Experiments show that this framework exhibits state-of-the-art performance. There is still some potential room for improvement of this framework, for example, evolutionary algorithms can be introduced to optimize the initial graph structure and improve the accuracy of the arithmetic model, which is currently being researched by our team in this direction and can be addressed in the future. The team goal is to further reduce the complexity of RA-HGNN, and a more exciting challenge is to apply the framework to directed graphs. In addition, there are different types of information in the real world, and attempting to broaden this method to handle multiple types of information like images.

CRediT authorship contribution statement

Zongxing Zhao: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft. **Zhaowei Liu:** Supervision, Writing – review & editing. **Yingjie Wang:** Project administration. **Dong Yang:** Resources. **Weishuai Che:** Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62272405, School and Locality Integration Development Project of Yantai City (2022), China, the Youth Innovation Science and Technology Support Program of Shandong Provincial under Grant 2021KJ080, the Natural Science Foundation of Shandong Province, China under Grant ZR2022MF238.

References

Atwood, J., & Towsley, D. (2016). Diffusion-convolutional neural networks. *Vol. 29*, In *Advances in neural information processing systems*. Curran Associates, Inc..

Cai, C., Wang, D., & Wang, Y. (2021). Graph coarsening with neural networks. In *International conference on learning representations*.

Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., et al. (2020). Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in Neural Information Processing Systems*, *33*, 17766–17778.

Choromanski, K. M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., et al. (2021). Rethinking attention with performers. In *International conference on learning representations*.

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*, *29*.

Ding, X., Xia, C., Zhang, X., Chu, X., Han, J., & Ding, G. (2021). Repmlp: Re-parameterizing convolutions into fully-connected layers for image recognition. arXiv preprint arXiv:2105.01883.

Dong, Y., Chawla, N. V., & Swami, A. (2017). Metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 135–144).

Fu, X., Zhang, J., Meng, Z., & King, I. (2020). MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of the web conference 2020* (pp. 2331–2341).

Grover, A., & Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 855–864).

Guan, C., Wang, X., & Zhu, W. (2021). Autoattend: Automated attention representation search. In *International conference on machine learning* (pp. 3864–3874). PMLR.

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Vol. 30*, In *Advances in neural information processing systems*. Curran Associates, Inc..

He, D., Liang, C., Huo, C., Feng, Z., Jin, D., Yang, L., et al. (2022). Analyzing heterogeneous networks with missing attributes by unsupervised contrastive learning. *IEEE Transactions on Neural Networks and Learning Systems*, 1–13.

Jia, Z., Lin, Y., Wang, J., Ning, X., He, Y., Zhou, R., et al. (2021). Multi-view spatial-temporal graph convolutional networks with domain generalization for sleep stage classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *29*, 1977–1986.

Jin, D., Huo, C., Liang, C., & Yang, L. (2021). Heterogeneous graph neural network via attribute completion. In *Proceedings of the web conference 2021* (pp. 391–400).

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Li, Z., Liu, H., Zhang, Z., Liu, T., & Xiong, N. N. (2022). Learning knowledge graph embedding with heterogeneous relation attention networks. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(8), 3961–3973.

Li, W., Ni, L., Wang, J., & Wang, C. (2022). Collaborative representation learning for nodes and relations via heterogeneous graph neural network. *Knowledge-Based Systems*, *255*, Article 109673.

Li, J., Peng, H., Cao, Y., Dou, Y., Zhang, H., Yu, P., et al. (2021). Higher-order attribute-enhancing heterogeneous graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 1.

Li, C., Yan, Y., Fu, J., Zhao, Z., & Zeng, Q. (2023). HetReGAT-FC: Heterogeneous residual graph attention network via feature completion. *Information Sciences*, *632*, 424–438.

Liu, C., Sun, L., Ao, X., Feng, J., He, Q., & Yang, H. (2021). Intention-aware heterogeneous graph attention networks for fraud transactions detection. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 3280–3288). Association for Computing Machinery.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, *234*, 11–26.

Liu, Z., Yang, D., Wang, Y., Lu, M., & Li, R. (2023). EGNN: Graph structure learning based on evolutionary computation helps more in graph neural networks. *Applied Soft Computing*, *135*.

Liu, Z., Yang, D., Wang, S., & Su, H. (2022). Adaptive multi-channel Bayesian graph attention network for IoT transaction security. *Digital Communications and Networks*.

Lu, G., Li, J., & Wei, J. (2022). Aspect sentiment analysis with heterogeneous graph neural networks. *Information Processing & Management*, *59*(4), Article 102953.

Luan, S., Hua, C., Lu, Q., Zhu, J., Zhao, M., Zhang, S., et al. (2022). Revisiting heterophily for graph neural networks. *Advances in Neural Information Processing Systems*, *35*, 1362–1375.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *ICLR*.

Park, N., Rossi, R. A., Ahmed, N., & Faloutsos, C. (2022). MetaGL: Evaluation-free selection of graph learning models via meta-learning. In *The eleventh international conference on learning representations*.

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 701–710).

Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., & Welling, M. (2018). Modeling relational data with graph convolutional networks. In *The semantic web* (pp. 593–607).

Sun, Y., Han, J., Yan, X., Yu, P. S., & Wu, T. (2011). PathSim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, *4*(11), 992–1003.

Suykens, J. A. (2001). Support vector machines: A nonlinear modelling and control perspective. *European Journal of Control*, *7*(2), 311–327.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015). LINE: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web* (pp. 1067–1077).

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *International conference on learning representations*.

Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., et al. (2019). Heterogeneous graph attention network. In *The world wide web conference* (pp. 2022–2032).

Wang, T., Xu, N., Chen, K., & Lin, W. (2021). End-to-end video instance segmentation via spatial-temporal graph neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10797–10806).

Wang, K., Yu, Y., Huang, C., Zhao, Z., & Dong, J. (2022). Heterogeneous graph neural network for attribute completion. *Knowledge-Based Systems*, *251*, Article 109171.

Wu, Z., Jain, P., Wright, M., Mirhoseini, A., Gonzalez, J. E., & Stoica, I. (2021). Representing long-range context for graph neural networks with global attention. *Vol. 34*, In *Advances in neural information processing systems* (pp. 13266–13279).

Xiang, C., Li, Y., Zhou, Y., He, S., Qu, Y., Li, Z., et al. (2022). A comparative approach to resurrecting the market of MOD vehicular crowdsensing. In *IEEE INFOCOM 2022 - IEEE conference on computer communications* (pp. 1479–1488).

Xiang, C., Zhang, Z., Qu, Y., Lu, D., Fan, X., Yang, P., et al. (2020). Edge computing-empowered large-scale traffic data recovery leveraging low-rank theory. *IEEE Transactions on Network Science and Engineering*, *7*(4), 2205–2218.

Xiao, H., Feng, J., Lin, G., Liu, Y., & Zhang, M. (2018). Monet: Deep motion exploitation for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1140–1148).

Xu, J., Dai, E., Zhang, X., & Wang, S. (2022). Hp-gmn: Graph memory networks for heterophilous graphs. In *2022 IEEE international conference on data mining (ICDM)* (pp. 1263–1268). IEEE.

Yan, W., Gu, M., Ren, J., Yue, G., Liu, Z., Xu, J., et al. (2023). Collaborative structure and feature learning for multi-view clustering. *Information Fusion*, *98*, Article 101832.

Yang, X., Yan, M., Pan, S., Ye, X., & Fan, D. (2023). Simple and efficient heterogeneous graph neural network. *Vol. 37*, In *Proceedings of the AAAI conference on artificial intelligence* (9), (pp. 10816–10824).

Zhang, B., Luo, S., Wang, L., & He, D. (2022). Rethinking the expressive power of GNNs via graph biconnectivity. In *The eleventh international conference on learning representations*.

Zhang, C., Song, D., Huang, C., Swami, A., & Chawla, N. V. (2019). Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 793–803).

Zhang, D., Yin, J., Zhu, X., & Zhang, C. (2018). MetaGraph2Vec: Complex semantic path augmented heterogeneous network embedding. In *Advances in knowledge discovery and data mining* (pp. 196–208).

Zhu, G., Zhu, Z., Wang, W., Xu, Z., Yuan, C., & Huang, Y. (2023). AutoAC: Towards automated attribute completion for heterogeneous graph neural network. arXiv preprint arXiv:2301.03049.