GGAT: Gravitation-Based Graph Attention Networks

Shujuan Wei, Huijun Tang, Pengfei Jiao, *Member, IEEE*, and Huaming Wu, *Senior Member, IEEE*

***Abstract*—Graph-structured data is an important data form that is widely used in the real world. It can effectively and abstractly express entities in information and the relationships between entities. The appearance of Graph Neural Networks (GNNs) provides a potent tool for dealing with nonlinear data structures, which mainly learns node representation through information propagation and aggregation on the nodes in the graph. However, existing GNNs fail to adequately and efficiently integrate the topological structure of the network and node features during information propagation, resulting in an insufficient capture of the complex influence relationships between nodes. The limitation constrains the expression ability of the models and seriously impacts their performance in node classification tasks. To overcome this issue, we propose a Gravitation-based Graph Attention Network (GGAT) for node classification. Firstly, we define a novel similarity measurement method based on the formula of universal gravitation, which combines node information entropy and spatial distance. This method overcomes the limitation of existing similarity measurements that focus solely on the topological structure or node features, achieving a more comprehensive similarity assessment. Then, we apply it to the graph attention network as a novel attention mechanism. Compared with the traditional attention mechanisms based on learning, our proposed mechanism not only thoroughly considers the topological structure and node features to allocate the weights of neighbor nodes but also makes the calculation of attention weights more transparent with an intuitive physical significance, thereby improving the stability and interpretability of the model. Finally, the experiments are carried out on various real datasets, and the results show that GGAT is superior to the existing popular models in node classification performance.**

***Index Terms*—Graph attention network, node gravitation, node information entropy, spatial distance.**

I. Introduction

As TECHNOLOGY improves by leaps and bounds, substantial data, which contains rich information and value, has been generated in various fields in the real world. Dealing with and digging out the hidden information in the data is a new challenge in the era of Big Data. Graph data with powerful expression ability can be used to model relationships between different entities in real-world scenes, in which many problems can be solved efficiently by mining and analyzing graph data [1]. For example, through modeling and analyzing the behaviors and interaction patterns of users in social networks, we can understand the needs of users and provide them with better services [2]. Each paper is regarded as a node and the citation relationship between papers is an edge. A citation network diagram is established, which can be analyzed to annotate and classify papers more quickly [3]. In the field of biomedicine, diseases, their similar causes and treatment methods can also be constructed as graph data to assist the treatment of diseases and design a more effective therapeutic regimen [4].

However, as a data structure in non-Euclidean space, graph data is difficult to be directly processed by traditional machine learning methods [5], [6]. The emergence of Graph Neural Network (GNN) and its variants [7], [8], [9] has settled this problem. GNNs combine graph data with neural networks, perform end-to-end calculations on graph data, and fuse node features and topological structure through information transmission and aggregation on nodes in the graph to obtain the representations of nodes [10], [11]. The feature information of a node determines the role and state of the node in the network, and the topological structure of the network determines how the information is propagated between nodes. Fully integrating the two is the key to improving the expression ability of GNNs and thus enhancing the performance of GNNs in node classification tasks.

Unfortunately, the early graph models [12], [13] are mostly based on random methods, which not only fail to directly use node features, but also lose part of the topological information in the graph, resulting in a limited model effect. Although the emergence of Graph Convolution Neural Networks (GCNs) [14], [15], [16] provides a brand-new perspective to learn graph information by convolution, their limitations are gradually emerging. GCNs assume that all neighbor nodes have equal influence on the central node, which ignores the heterogeneity between neighbor nodes. It leads to the fact that when a node has the special feature or occupies an important position in the topological structure, the model still treats it as equivalent to other nodes, and thus fails to fully capture the impact of the difference between different nodes, which limits its expression ability.

Attention-based GNNs (GATs) [17], [18], [19] overcome the limitation of GCNs by introducing an attention mechanism that can assign weights to different neighbors. However, their methods of weight calculation primarily rely on node features, utilizing adjacency relationships merely to constrain the scope of weight calculation, without directly modeling the topological structure information. These calculation methods lead the models to ignore the role of topological information in weight

Received 26 October 2024; revised 30 March 2025; accepted 19 June 2025. Date of publication 26 June 2025; date of current version 8 July 2025. This work was supported in part by the National Key R&D Program of China under Grant 31400 and in part by the National Natural Science Foundation of China under Grant 62071327. The associate editor coordinating the review of this article and approving it for publication was Dr. Elif Vural. *(Corresponding author: Huaming Wu.)*

Shujuan Wei and Huaming Wu are with the Center for Applied Mathematics, Tianjin University, Tianjin 300072, China (e-mail: weishujuan@tju.edu.cn; whming@tju.edu.cn).

Huijun Tang and Pengfei Jiao are with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: tanghuijune@hdu.edu.cn; pjiao@hdu.edu.cn).

Digital Object Identifier 10.1109/TSIPN.2025.3583355

allocation, which makes it challenging for the models to effectively describe the topological structure of the network. Recent studies [20], [21] have attempted to incorporate the topological information when computing neighbor weights. Nevertheless, these methods usually learn weights separately based on the topological structure and node features, and then integrate them in a simple way, which regards the topological structure and feature information as two completely independent parts, neglecting the potential relationship between them and preventing the model from fully leveraging both pieces of information.

Moreover, most GATs generate weights between nodes by learning a trainable weight matrix. The learning-based attention mechanism relies excessively on training data, which can lead to unstable training processes in the face of insufficient training data or complex relationships between nodes. Additionally, the learned weights lack the intuitive physical explanation, rendering the decision-making process of the model difficult to understand.

To solve the above problems, we propose a Gravitation-based Graph Attention Network (GGAT). Specifically, we design a novel attention mechanism based on node similarity, aiming to improve the traditional models with a more rational and intuitive weight allocation strategy. Our fundamental hypothesis is that the more similar the neighbor node is to the central node, the more important it is to the central node. For instance, in an academic citation network, papers sharing the same topic as the target paper often carry greater reference value. Therefore, we take the similarity between nodes as the core basis for weight allocation.

Regarding the selection of the similarity measurement, we find that traditional methods generally exhibit limitations. For example, Cosine similarity is based on the angle between node features, ignoring the topological structure information, while Jaccard similarity is based on the number of common neighbors, without incorporating node features. To overcome these shortcomings, we aim to define a node similarity measurement that comprehensively considers both the topological structure and node features. Inspired by the law of universal gravitation, which states that the gravitational force between two objects is directly proportional to the product of their masses and inversely proportional to the square of the distance between them, we construct a novel similarity measurement. We regard the similarity between nodes as a form of "node gravitation", which is influenced both by the inherent importance of the nodes in the network structure and their proximity in the feature space, that is, the more important the nodes are within the network structure and the closer the nodes are in the feature space, the greater the node gravitation between them.

Specifically, we employ node information entropy to characterize the importance of a node within the network structure: node information entropy quantifies the structural information contained in a node by considering the degrees of the node and its neighbors. A greater information entropy indicates that the node possesses richer information and a more complex topological structure [22]. Simultaneously, we use spatial distance to measure the proximity of nodes in the feature space: a smaller distance indicates greater similarity in node features. By adopting a product coupling method to nonlinearly integrate the two aspects, we propose a node similarity measurement based on the formula of universal gravitation, which more comprehensively reflects the matching degree of nodes in structure and features, and realizes the synergistic mechanism between the structural information importance of nodes and the proximity of the feature space. The main contributions of this paper can be summarized as follows:

- We define a novel method to measure the similarity of nodes based on the formula of universal gravitation, which breaks through the limitation of traditional similarity measurements that tend to focus solely on either the topological structure or node features, and achieve a more thorough node similarity measurement by fusing the topological structure information of nodes in the network and the feature expression of nodes.
- We assume that the neighbor node with stronger gravitation to the central node is more similar and thus relatively more important. By taking node gravitation as the weight coefficient of each neighbor node, we propose a novel attention mechanism based on node gravitation, which can assign weights to the neighbors in a more physically meaningful way that incorporates the topological structure and node features, rendering the calculation of weights more transparent and enhancing the stability and interpretability of the model.
- We apply the attention mechanism based on node gravitation to GAT, and propose a novel graph attention network framework based on node gravitation. Comparative experiments on different datasets with existing node classification works prove the effectiveness of our scheme.

The rest of the paper is as follows. Section II introduces the related research on node classification tasks. In Section III, we present the key technologies and symbols used in this model. In Section IV, we put forward GGAT and analyze the key parts. We carry out extensive experiments to verify the superiority of the proposed model in Section V, and make a summary of the paper in Section VI.

II. Related Work

With the continuous advancement of GNNs, various models have been proposed to improve the accuracy of node classification tasks [23], [24], [25].

Brua et al. [26] presented a Spectral Convolution Neural Network, which realizes node classification by performing eigenvalue decomposition of the Laplacian matrix of the graph and then defining convolution operations in the Fourier domain. However, the method is costly in terms of non-spatial localization and intensive calculation. Defferrard et al. [27] used the Chebyshev polynomial instead of the convolution kernel on the basis of the above, eliminating the process of calculating the Laplacian matrix and effectively reducing the computational complexity of the graph convolution process. Kipf et al. [14] improved the Chebyshev network, further diminished the computational complexity of the graph convolution process, and proposed the Graph Convolution Network (GCNs).

The aforementioned network models are all GCNs defined in the spectral domain. Hamilton et al. [28] designed a graph sampling and aggregation model from the node domain, which resolves the problem of memory explosion in GCN. It represents new feature information by randomly sampling and aggregating neighbor nodes instead of a graph Laplacian matrix. Rong et al. [29] randomly removed a certain number of edges from the input graph in each training period, so as to reduce the convergence speed of over-smoothness and the information loss caused by it. Wang et al. [30] showed that the most advanced GCNs are far from satisfactory in fusing the topological structure and node features, and designed an Adaptive Multi-channel Graph Convolutional Networks (AM-GCN) for semi-supervised classification. Yang et al. [31] dynamically adjusted the adjacency relationships of the graph by establishing a soft association between node attributes and topological structure to improve the node classification performance of GNNs on homogenous and heterogeneous graphs.

The above network models ignore that different nodes have different importance to the central node, and thus limit their own expression ability and the final classification performance. To address the problem, some works have attempted to improve traditional GCNs by introducing attention strategies. Velickovic et al. [17] considered that each neighbor node had a different contribution to the central node, and then introduced the attention mechanism to define the aggregation function. Hao et al. [32] proposed the Neighbor Enhanced Graph Convolution Network (NEGCN) that introduces an efficient edge classifier to explicitly identify useful neighbors. Peng et al. [33] adopted a label-aware neural similarity measurement based on node attributes to determine the most similar adjacency point, which can avoid over-embedding among various types of nodes, and proposed an enhanced relationship-aware neighbor mechanism to select the most similar adjacency point of the target node before aggregating all the adjacency point information in different relationships. Wang et al. [34] put forward the Graph Quaternion-Valued Attention Network (GQAT) for node classification, which replaces vector multiplication in self-attention with quaternion vector multiplication and uses shared parameters to reduce the model parameters. Kim et al. [35] put forward an improved graph attention model for noise graphs, which uses two attention tables compatible with self-supervised tasks to predict edges, and through coding edges, learns more attention when distinguishing wrongly linked neighbors. Peng et al. [36] proposed a novel model, which defines the topology and evaluates node importance by the degrees of nodes, and combines the self-attention mechanism to capture the relationships between the central node and its neighbors, so as to achieve a more accurate node classification in complex directed graphs. Chen et al. [20] introduced the Content- and Structure-based Graph Attention Network (CSGAT), which separately generates content coefficients from node features and structure coefficients from the adjacency matrix, and then integrates the two types of coefficients to form the attention weights, which fully considers both the node content similarity and the structural similarity.

Although such models can assign different weights to neighbor nodes through the attention mechanism, their weight calculation processes either rely solely on node features without directly incorporating the topological structure or respectively learn independent attention weights for the topological structure and node features, followed by a simple fusion. This hinders the effective integration of network topology and node features, and constrains the models' ability to extract hidden inter-node relationships. In terms of the problem, we present a novel GAT framework based on node gravitation, which can make full use of the topological structure information in the network and feature expression of nodes through node information entropy and spatial distance between nodes, catch the hidden complex relationships in the network, and then more efficiently complete the node classification tasks.

TABLE I
SYMBOLS AND EXPLANATIONS

Symbol	Explanation
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Graph
$\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$	Set of nodes
$\mathcal{E}$	Set of edges
N	Number of nodes
C	Number of edges
$\hat{N}_i$	First-order neighbors set of node v_i
$\hat{D} = \{\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_N\}$	Degree matrix
$A \in R^{N \times N}$	Adjacency matrix
$X = \{x_1, x_2, \ldots, x_N\}$	Node feature matrix
Q and Q'	Number of features
a_{ij}	Attention coefficient between node v_i and v_j
$H(Y)$	Information entropy of the random variable Y
$F(\cdot, \cdot)$	Node gravitation function
E_i	Information entropy of node v_i
$d(\cdot, \cdot)$	Spatial distance function
$h = \left\{\vec{h_1}, \vec{h_2}, \ldots, \vec{h_N} \mid \vec{h_i} \in R^Q\right\}$	Node features of input layer
$h' = \left\{\vec{h'_1}, \vec{h'_2}, \ldots, \vec{h'_N} \mid \vec{h'_i} \in R^{Q'}\right\}$	Node features of output layer
K	Number of attention heads

III. PRELIMINARY

An input graph can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ respectively denote the set of N nodes and C edges. For a node $v_i \in \mathcal{V}$, $e_{ij} = (v_i, v_j) \in \mathcal{E}$ denotes an edge, where v_i and v_j respectively represent the two endpoints of e_{ij}. $\hat{N}_i$ and $\hat{d}_i$ are denoted as the set of the first-order neighbors of v_i and the degree of v_i respectively. The adjacency matrix is defined as $A \in R^{N \times N}$, where $A(i, j) = 1$ indicates that there is an edge between nodes v_i and v_j. Let $X \in R^{N \times Q}$ represent the node feature matrix, where Q denotes the number of features for each node. The symbols and explanations in the paper are summarized in Table I.

A. Graph Attention Network

GAT [17] applies the attention mechanism in deep learning to GNNs, using the self-graph attention layers to assign different weights to different nodes in the neighborhood. The method not only avoids complicated matrix operations, but also does not need to know the structure of the graph in advance, and has achieved excellent performance in the tasks of node classification. The calculation coefficient of the attention mechanism is:

$$a_{ij} = \frac{\exp\left(LeakyRELU(\vec{a}^T[W\vec{h_i}||W\vec{h_j}]\right)}{\sum_{b \in \hat{N}_i} \exp\left(LeakyRELU(\vec{a}^T[W\vec{h_i}||W\vec{h_b}|)\right)}, \quad (1)$$

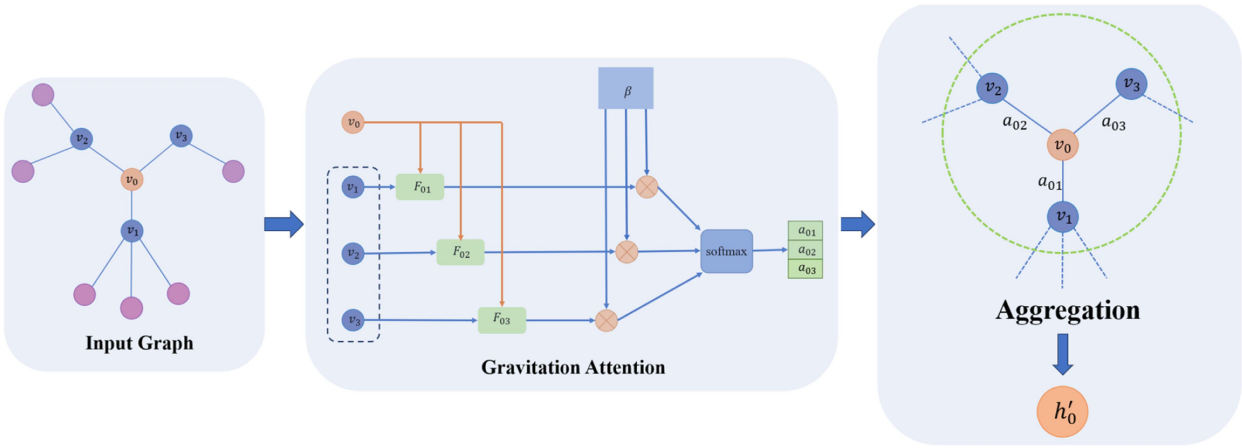


Fig. 1. The framework of GGAT. Given an input graph, where the orange node indicates the central node and the blue nodes denote the aggregated neighbors, the gravitation attention module operates as follows: firstly, node gravitation F_{ij} between the central node and each neighbor is computed, and then F_{ij} is normalized to obtain the attention coefficient a_{ij} between nodes. Finally, the neighbors are aggregated through a weighted sum based on a_{ij} to get the final embedding h'_i of the central node v_i.

where $||$ is the concatenation operation and $\hat{N}_i$ represents the set of direct neighbors of the node v_i in the graph. W and $\vec{a}$ are parameters that need to be tuned in training. a_{ij} denotes the weight between node v_i and node v_j. The node feature of the input layer is defined as $h = \{\vec{h_1}\ \vec{h_2}\ \ldots, \vec{h_N} \mid \vec{h_i} \in R^Q\}$, where N denotes the number of nodes and Q represents the dimension of features.

B. Information Entropy

Information entropy is a concept in information theory that is used to denote the uncertainty and randomness of information and quantify the amount of information in an information source. Assuming that $Y = \{y_1, y_2, \ldots, y_t\}$ has t different values, and the probability of each value is $P(y_i)$, then the information entropy $H(Y)$ of Y can be expressed as:

$$H(Y) = -\sum_{i=1}^{t} P(y_i) \log P(y_i). \tag{2}$$

The higher the information entropy, the greater the uncertainty of the information source, and the greater the expected value of the information of all possible events.

C. Universal Gravitation

Gravitation is an interaction between objects whose magnitude is related to the masses of the objects and the distance between objects. When the masses of objects are greater and the distance between two objects is closer, the gravitational force between them is greater. The gravitation between two objects can be regarded as particles, which can be calculated by the following formula:

$$F = \frac{GmM}{r^2}, \tag{3}$$

where m and M are the masses of particles, r is the distance between them, and G represents the gravitational constant.

IV. Gravitation-Based Graph Attention Network

In this paper, we propose the Gravitation-based Graph Attention Network, which involves three steps. Firstly, based on node information entropy and spatial distance, referring to the gravitation formula, we define the node gravitation to measure the similarity between nodes. Then, the node gravitation is equivalent to the weight coefficient of each neighbor node to get an attention mechanism based on node gravitation. Finally, we gain a novel graph attention network model by applying it to GAT. The framework of GGAT is shown in Fig. 1.

A. Similarity Measurement Based on Universal Gravitation

We regard the nodes in the graph as the particles, equate the information entropy of nodes with the masses of particles, and take the spatial distance of the nodes as the distance between particles, and figure the similarity between nodes according to the formula of universal gravitation. The schematic diagram is shown in Fig. 2. The advantage of the method is that it nonlinearly incorporates both node information entropy and spatial distance in a product coupling way, and realizes the synergistic mechanism between the structural information importance of nodes and the proximity of the feature space. Compared with existing similarity measurements, it more fully considers the topological structure information and feature expression of nodes, which can better evaluate the similarity between nodes.

1) Node Information Entropy: When measuring the role played by a node in the network, node information entropy can be used to describe the richness of information carried by the node and its impact on the transmission of network information. It is a great way to assess node information [22], [37], which takes into account the degrees of the central node and its neighbors. Differing from the methods that only use the degree of the node itself, it can use the combined degree information from the node and its neighbors to evaluate the topological importance more comprehensively [38]. The greater the information entropy of a node, the richer the information it contains in the network

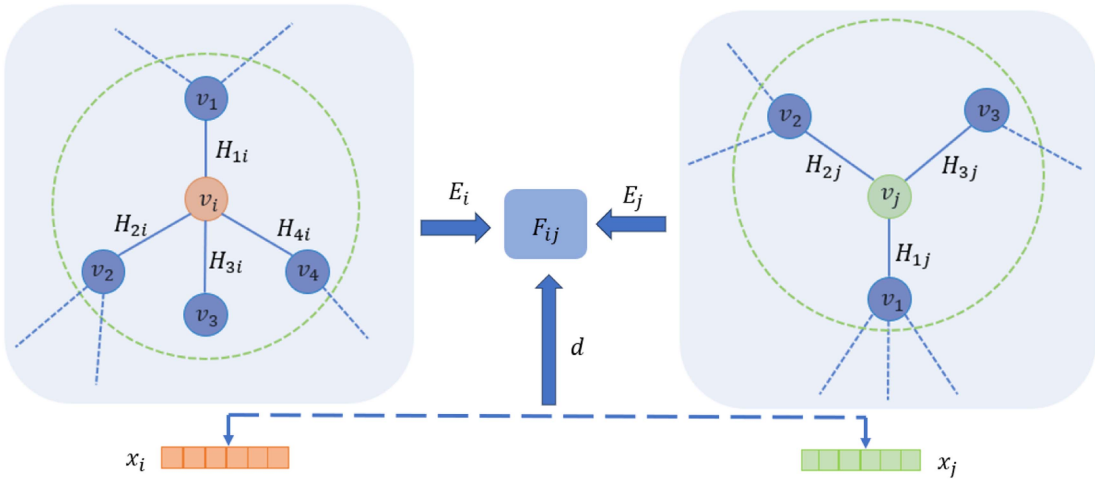


Fig. 2. The diagram of node gravitation, where H_{ji} denotes the influence of node v_j on node v_i. x_i and E_i, respectively, represent the feature and information entropy of v_i. d and F_{ij} are the spatial distance and node gravitation between v_i and v_j.

structure, and thus the more important it is in the network. Moreover, in the gravitation model, the mass of a node is usually reflected in its importance or influence in the network [39], [40], so we equate the node information entropy with the mass of a particle, and the greater the node information entropy, the greater the node mass.The information entropy of each node v_i is as follows:

$$E_i = \sum_{j \in \hat{N}_i} H_{ji} = \sum_{j \in \hat{N}_i} p_{ji} \log \frac{1}{p_{ji}}, \tag{4}$$

where $p_{ji} = \frac{\hat{d}_j}{\sum_{l \in \hat{N}_i} \hat{d}_l}$, $\hat{N}_i$ is indicated as the first-order neighbor of v_i, $\hat{d}_j$ is denoted as the degree of v_j, and $\sum_{l \in \hat{N}_i} p_{li} = 1$.

2) Spatial Distance: Distance measurement is utilized to standardize the distance between nodes in the feature space, that is, the proximity of nodes in the feature space. we equate the spatial distance between node features with the distance between particles. The greater the distance, the smaller the node gravitation and the weaker the influence. Taking Euclidean distance as an example, the distance between node v_i and node v_j can be described as:

$$d(v_i, v_j) = \sqrt{\sum_{s=1}^{q} (x_{is} - x_{js})^2}, \tag{5}$$

where $x_i = \{x_{i1}, x_{i2}, \ldots, x_{iq}\}$ is the features of v_i, and $x_j = \{x_{j1}, x_{j2}, \ldots, x_{jq}\}$ is the features of v_j.

3) Node Gravitation: Node gravitation is influenced by the importance of nodes in the network structure and their proximity in the feature space. According to (3), we define the node gravitation for measuring node similarity in graph data as follows:

$$F(v_i, v_j) = G_n \frac{E_i E_j}{d(v_i, v_j)^{\alpha}}, \tag{6}$$

where G_n is the node gravitation constant and α is a parameter that adjusts the sensitivity of the agent to changes in the performance of the model.

B. Attention Mechanism Based on Node Gravitation

We suppose that the more potent the gravitation between nodes, the higher their similarity, i.e., the neighbor nodes with stronger gravitation to the central node are relatively more important, and then assign them higher weights. Therefore, we directly consider the node gravitation obtained in (6) as the weight coefficient of each neighbor node after normalization operation, i.e., during the training of GAT, we utilize the attention mechanism based on the similarity measurement of node gravitation instead of the traditional attention mechanism for learning attention weight. This attention mechanism can assign weights to the neighbors in a more physically meaningful way that incorporates the topological structure and node features, increases the transparency of the model, and makes the decision-making process of the model easier to understand. The attention coefficient can be calculated as:

$$a_{ij} = \frac{\exp\left(\beta \cdot F(W\vec{h_i}, W\vec{h_j})\right)}{\sum_{b \in \hat{N}_i} \exp\left(\beta \cdot F(W\vec{h_i}, W\vec{h_b})\right)}, \tag{7}$$

where a_{ij} is the weight between node v_i and node v_j, W is a learnable weight matrix and β is a learnable bias.

C. Feature Updating

After the relative importance weight corresponding to each neighbor node of the central node is calculated by (7), the features of neighbor nodes are weighted and summed by the nonlinear activation function σ in accordance with the message passing mechanism to update its feature.

$$\vec{h_i'} = \sigma\left(\sum_{j \in \hat{N}_i} a_{ij} W \vec{h_j}\right), \tag{8}$$

where $h = \{\vec{h_1}, \vec{h_2}, \ldots, \vec{h_N} \mid \vec{h_i} \in R^Q\}$ is the node features of input layer, and $h' = \{\vec{h_1'}, \vec{h_2'}, \ldots, \vec{h_N'} \mid \vec{h_i'} \in R^{Q'}\}$ is the node features of output layer. N denotes the number of nodes, Q and Q' represent the dimension of features.

Algorithm 1: GGAT: Gravitation-Based Graph Attention Network.

Require: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, parameter α, attention head K, node feature $h = \{\vec{h_1}\ \vec{h_2}\ \ldots, \vec{h_N} \mid \vec{h_i} \in R^Q\}$

Ensure: feature matrix $h' = \{\vec{h'_1}, \vec{h'_2}, \ldots, \vec{h'_N} \mid \vec{h'_i} \in R^{Q'}\}$

```
Initialize weight matrix W, bias β and node feature h
for v_i ∈ V do
  for (v_i, v_j) ∈ E do
    p_ji ← d̂_j / Σ_{l∈N̂_i} d̂_l
    E_i ← (4)
  end for
end for
for v_i ∈ V do
  for k = 1 → K do
    for (v_i, v_j) ∈ E do
      d(v_i, v_j) ← (5)
      F(v_i, v_j) ← (6)
      a_ij ← (7)
    end for
    if the result of the hidden layer then
      h'_i ← (9)
    else
      h'_i ← (10)
    end if
  end for
end for
```

To further upgrade the expression ability of the attention layer, we adopt the multi-head attention mechanism, which adds several groups of independent attention, so that it can allocate attention to multiple related features between the central node and its neighbor nodes, and thus enhance the stability of the node embedding vector. We invoke K sets of independent attention mechanisms on (8). For the output result of the hidden layer, we concatenate the embedding vectors of node v_i learned by the multi-head attention mechanism in (9). For the result of the output layer, we average the embedding vectors of node v_i to form the final node embedding vector in (10).

$$\vec{h'_i} = \|_{k=1}^{K} \sigma\left(\sum_{j \in \hat{N}_i} a_{ij}^k W^k \vec{h_j}\right), \tag{9}$$

$$\vec{h'_i} = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in \hat{N}_i} a_{ij}^k W^k \vec{h_j}\right), \tag{10}$$

where $||$ is the concatenation operation and a_{ij}^k represents the normalized attention coefficient calculated by the k-th attention mechanism. W^k denotes the weight matrix of the corresponding input linear transformation, and σ is the final nonlinear function, such as softmax. The overall process of GGAT is detailed in Algorithm 1.

D. Complexity Analysis

Our proposed GGAT demands to calculate the feature mapping of all nodes and node gravitation between nodes and their neighbors, so the time complexity of the single-head GGAT model is $\mathcal{O}(NQQ' + CQ')$, where N and C respectively indicate the number of nodes and edges, Q and Q' denote the dimension of input feature and output feature. The attention mechanism of multiple heads is K times that of a single head, where K is the number of attention heads.

V. Experiment

This section first introduces the datasets used in the experiment and relevant experimental settings, and then compares our model with popular GNNs for node classification to demonstrate its superiority. Finally, we examine the influence of different distances on the model and analyze the parameter sensitivity.

A. Experimental Setting

1) Datasets: We conduct experiments based on the following six real datasets.

- Cora, Citeseer, Pubmed [14]:[1] These are standard citation network datasets, which are composed of nodes representing papers and edges representing the citation relationships between two papers. Nodes are characterized by the word bag representations of papers, and node labels are the academic topics of papers. All nodes are divided into 7, 6, and 3 categories, respectively.
- ACM [41]:[2] This is an academic network dataset that records the publication of scholars' literature. Taking the papers as nodes, the research directions of papers are divided into three categories according to the nature of the conferences in which the papers are published. The initial feature of each paper node is expressed by the bag vector of the paper term.
- CoraFull [42]:[3] This is a larger version of the above Cora dataset, which is reported by most approaches, and we name it CoraFull. As Cora, the dataset represents papers as nodes, citation relationships between papers as edges, and in particular, all of its nodes are divided into 70 classes.
- DBLP [42]:[4] This is a large computer science literature database, which is mainly used to store information such as academic papers, conference papers and books, and it consists of core elements such as nodes, edges and features, where all nodes are divided into four categories.

To be specific, we randomly select 20 nodes in the peer class of datasets as the training set, and randomly select 1000 nodes as the testing set. Details of the datasets are shown in Table II:

2) Baselines: Our proposed GGAT is compared with nine baselines which are listed as follows. Among them, GCN, GraphSAGE, and GAT are classical GNN models, GATv2,

[1]https://github.com/tkipf/gcn
[2]https://github.com/Jhy1993/HAN
[3]https://github.com/abojchevski/graph2gauss/
[4]https://github.com/abojchevski/graph2gauss/

TABLE II
THE CHARACTERISTICS OF THE DATASETS

Dataset	Nodes	Edges	Features	Classes	Training nodes	Testing nodes
Cora	2708	5429	1433	7	140	1000
Citeseer	3327	44338	3703	6	120	1000
ACM	3025	13128	1870	3	60	1000
Pumbed	19717	81894	500	3	60	1000
CoraFull	19793	65311	8710	70	1400	1000
DBLP	17716	105734	1639	4	80	1000

CAT, Simp-GCN, FastGAT, CSGAT, and GTAT are relatively advanced GNN models.

- **GCN [14]**: It applies the trainable convolution operation to graph data. Based on the first-order approximation of spectral convolution on the graph, an effective hierarchical propagation mechanism is designed to extract graph structure data features.
- **GraphSAGE [28]**: It directly uses the topological structure of the graph, uniformly and randomly samples the neighbor nodes of each node, and then performs specific aggregation operations in the field to update the feature of the current central node.
- **GAT [17]**: It introduces the attention mechanism in deep learning into GNNs for the first time, and generates the node representation by assigning different weight coefficients to the neighbor nodes of each node.
- **GATv2 [43]**: It is an improved version of the original GAT, which allows the network to dynamically gather different nodes when processing graph data. By introducing a more powerful attention mechanism, the ability to express and perform the network is optimized.
- **CAT [44]**: It is a graph joint attention network, which flexibly combines node features with learnable structural interventions outside the GNN to calculate more appropriate attention coefficients, and thereby enhances the learning capability of the model.
- **SimP-GCN [45]**: It integrates the graph structure and node features in a data-driven way, so that each node can adaptively adjust the information from both of them, and applies the self-supervised strategy to explicitly capture the similar relationships between features.
- **FastGAT [19]**: It proposes a graph attention network that can globally perceive and adaptively calculate the attention of nodes, which not only improves the calculation efficiency, but also enhances the potency of the model.
- **CSGAT [20]**: It introduces a hybrid attention mechanism that integrates node content and structural information, enables more accurate attention weight estimation between nodes, and enhances the convergence of the model.
- **GTAT [21]**: It uses the cross-attention mechanism to fuse the original feature representation and topological representation, so that the model can adjust the influence of node features and topological structure, and enhance the expression ability of the model.

3) Parameters Setting: To more comprehensively assess our model, the parameters of all baselines are initialized with the same values as the original papers to reduce the impact of other factors, and we also further carefully adjust parameters to achieve better results. All the compared baselines set the learning rate to 0.01 and the L2 regularization to 5e-4. The dropout rates of GCN, GraphSAGE, and Simp-GCN are set to 0.5 and the rest are 0.6. The attention heads of the graph attention network models are 8 and the number of features calculated per attention head is 8. The output attention heads are 8 on the Pumbed and the other datasets are 1. The parameter details of the proposed GGAT are summarized in Table III and the GPU used is NVIDIA GeForce RTX 4060.

TABLE III
PARAMETER SETTINGS OF GGAT

Dataset	Dropout	Learning rate	Weight-decay	α
Cora	0.6	0.005	1e-4	1.5
Citeseer	0.6	0.01	5e-4	1.5
ACM	0.6	0.005	5e-4	1.5
Pumbed	0.6	0.005	1e-3	1.5
CoraFull	0.6	0.01	5e-4	1.5
DBLP	0.6	0.01	5e-4	1.5

TABLE IV
NODE CLASSIFICATION ACCURACY (%)

Method	Cora	Citeseer	ACM	Pubmed	CoraFull	DBLP
GCN	81.3	70.8	87.2	79.4	54.7	75.6
GraphSAGE	80.2	71.3	86.7	78.7	55.0	75.3
GAT	83.2	71.9	88.1	79.8	56.1	76.7
GATv2	80.9	71.6	86.4	78.5	52.3	72.2
CAT	82.5	72.1	88.6	81.0	53.7	76.1
Simp-GCN	83.0	72.6	89.1	80.6	53.9	76.5
FastGAT	83.5	**72.9**	88.4	79.9	55.3	76.2
CSGAT	82.9	72.1	88.7	79.9	51.6	75.5
GTAT	83.3	71.9	89.0	80.3	54.8	77.0
GGAT	**84.6**	72.4	**89.8**	**81.4**	55.8	**77.6**
GAT-J	83.5	70.7	88.3	80.2	**56.4**	75.9
GAT-A	77.4	66.0	88.2	78.7	49.6	74.8
GAT-C	83.1	70.6	87.9	79.6	52.6	75.2

Bold: best.

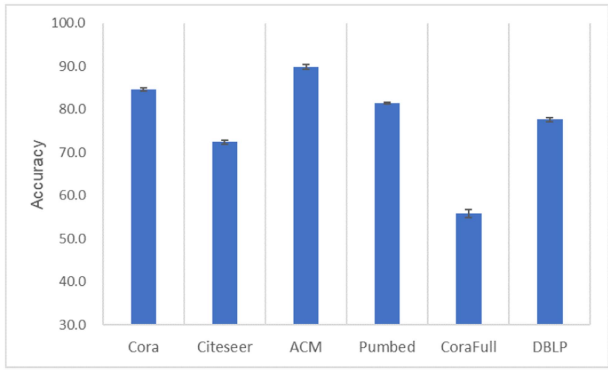


Fig. 3. The node classification accuracy and error bars (%) of GGAT.

B. Result Comparison

1) Cross-Dataset Evaluation of Classification Accuracy: We run 10 times for each dataset to get the average classification accuracy. The accuracy of nine baseline models and the proposed GGAT across six datasets is presented in Table IV. From the experimental results, it can be concluded that GGAT is optimal on Cora, ACM, Pumbed and DBLP, and its accuracy is increased by 1.1%, 0.7%, 0.4% and 0.6% in comparison with the optimal method in the benchmark models, respectively. The sub-optimal

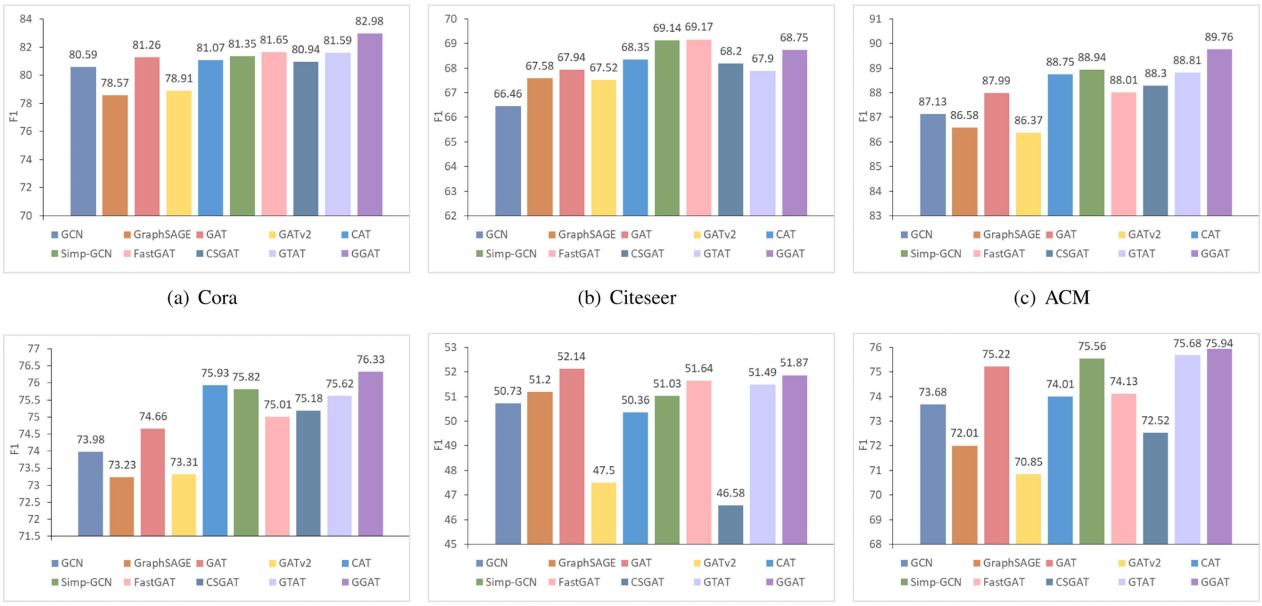


Fig. 4. The macro F1-score (%) of models on datasets.

result is also achieved on CoraFull and the performance on Citeseer is slightly inferior. We use error bars to represent the standard deviation and the results are shown in Fig. 3.

2) Comparative Study of Similarity Measurement Methods: To verify the advantages of the approach based on node gravitation as the similarity measurement compared with other similarity measurements, we use Jaccard, Cosine, and Adamic-Adar instead of the learning attention weight in GAT to obtain three GAT variants, namely, GAT-J, GAT-C, and GAT-A, respectively. Compared GGAT with the above three variant models on six datasets, it can be seen from Table IV that, except for the CoraFull, the accuracy of the proposed GGAT model is higher than that of the three GAT variants. It is apparent that the method adopting node gravitation as the similarity measurement is more efficient than the other three commonly used similarity measurements.

3) Model Performance Assessment With Macro F1-Score: To more comprehensively measure the performance of models, we also utilize macro F1-score (F1) as the evaluation criterion. The macro F1-score of each model on six datasets is shown in Fig. 4. GGAT outperforms other methods on four datasets (excluding Citeseer and CoraFull), while still obtaining results close to the top-performing methods on Citeseer and CoraFull. By comparing the accuracy and macro F1-score, it can be seen that our model can more fully extract the complex influence relationships between nodes, and boost the efficiency of node classification.

C. Analysis of Variants

We discuss the effect of node gravitation, node information entropy and spatial distance as attention coefficients on the model performance, respectively, in which the selection of spatial distance is consistent with node gravitation.

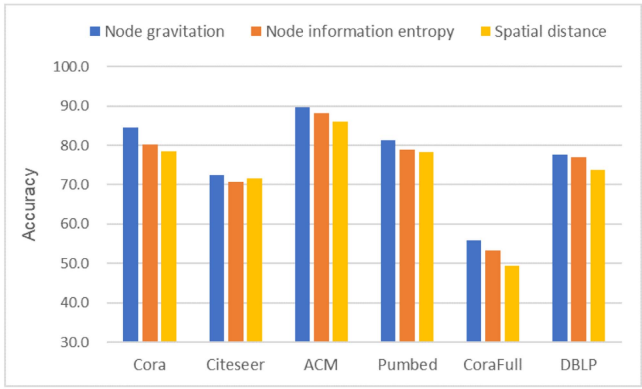


Fig. 5. The effect of GGAT variants on the accuracy (%) of model.

It can be seen from Fig. 5 that the accuracy of taking node gravitation as the weight coefficient of the attention mechanism on the six datasets is always higher than the other two cases, indicating that node gravitation utilizes node information entropy and spatial distance simultaneously, perfectly considers the topological structure information and node feature information, and improves the expression ability of the model.

D. Sensitivity Test on Spatial Distance and α

In this section, we discuss the influence of different spatial distances and hyperparameter α on the accuracy of node classification.

In machine learning, distance is used to assess the relationship between two objects. Determining an appropriate distance measurement will have an incredible impact on the effect of the whole learning algorithm. Let coordinates of points $\hat{X}$ and $\hat{Y}$ in q-dimensional space be $\hat{X} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_q\}$ and

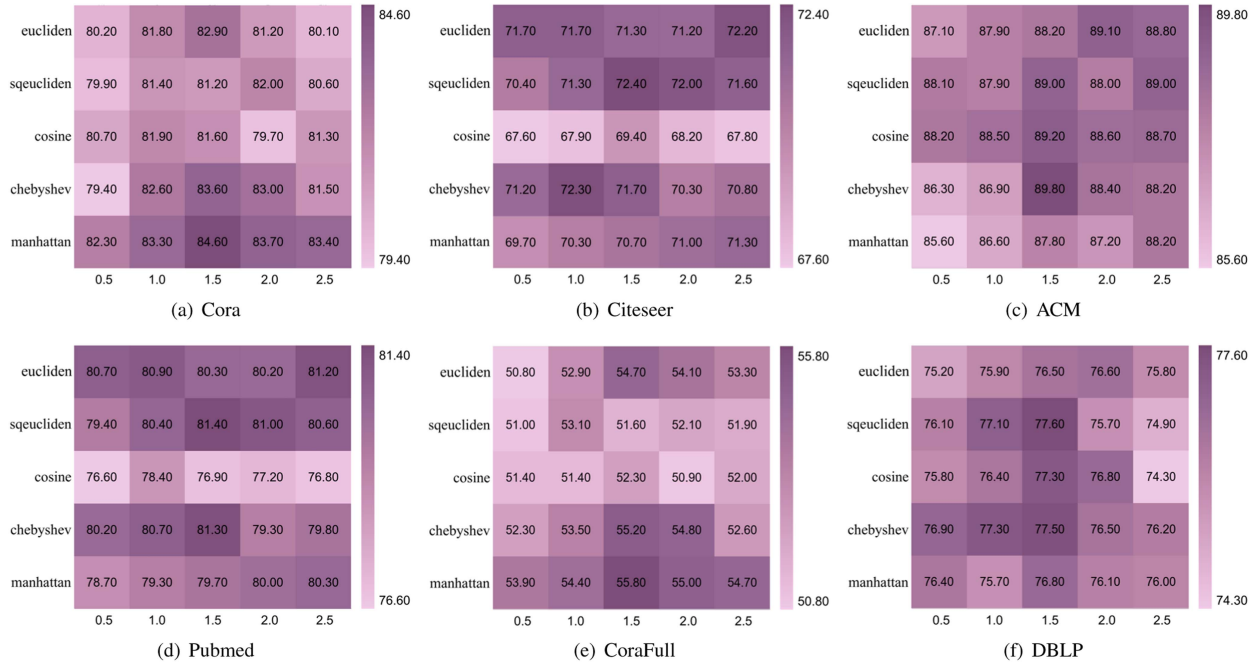


Fig. 6. The effect of space distances and hyperparameter α on the accuracy (%) of GGAT.

$\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_q\}$ respectively. There are many methods to calculate the distance between $\hat{X}$ and $\hat{Y}$, mainly as follows:

- **Euclidean** is the most common way to measure distance, which is to measure the absolute distance between two points in multidimensional space. The Euclidean distance between them is

$$d(\hat{X}, \hat{Y}) = \sqrt{\sum_{i=1}^{q} (\hat{x}_i - \hat{y}_i)^2}. \tag{11}$$

- **Sqeuclidean** is an improvement of Euclidean distance. All dimensions of data are standardized to mean and variance, and then brought into the calculation formula of Euclidean distance. The standardized Euclidean distance expression of $\hat{X}$ and $\hat{Y}$ is as follows:

$$d(\hat{X}, \hat{Y}) = \sqrt{\sum_{i=1}^{q} \left(\frac{\hat{x}_i - \hat{y}_i}{s_i} \right)^2}, \tag{12}$$

where s_i is the standard deviation of the component.

- **Cosine** refers to the cosine value of the angle between two vectors in a vector space, which is also called cosine similarity. Their cosine distance is defined as follows:

$$d(\hat{X}, \hat{Y}) = \frac{\sum_{i=1}^{q} \hat{x}_i \hat{y}_i}{\sqrt{\sum_{i=1}^{q} \hat{x}_i^2} \sqrt{\sum_{i=1}^{q} \hat{y}_i^2}}. \tag{13}$$

- **Chebyshev** is also known as infinite norm distance, which is a method to measure the distance between two vectors according to the maximum difference between them. Chebyshev distance in q-dimensional space can be expressed as:

$$d(\hat{X}, \hat{Y}) = \max_i(|\hat{x}_i - \hat{y}_i|). \tag{14}$$

- **Manhattan** is a geometric term used in geometric metric space to indicate the sum of the absolute distances between two points in a standard coordinate system. Manhattan distance of $\hat{X}$ and $\hat{Y}$ is defined as follows:

$$d(\hat{X}, \hat{Y}) = \sum_{i=1}^{q} |\hat{x}_i - \hat{y}_i|. \tag{15}$$

We choose α from 0.5 to 2.5, and the spatial distance takes the five distances mentioned above to carry out the experiment. According to the experimental results shown in Fig. 6, we can see that different parameters have a profound impact on the model performance. The trend of the influence of different spatial distances on the performance of the model varies across different datasets, which is mainly due to the difference in the number of nodes and their distribution in the network in each dataset. Cora and CoraFull reach the optimal values on Manhattan, Citeseer, Pubmed and DBLP reach the optimal values on Sqeuclidean, and ACM reaches the optimal value on Chebyshev. For parameter α, the accuracy of the six datasets shows a trend of increasing first and then decreasing with α at their optimal distance, and the best result is achieved when $\alpha = 1.5$.

VI. Conclusion

In this paper, a novel node classification model is designed to address the problem that the existing GNNs fail to sufficiently and efficiently integrate the topological structure information and feature information of nodes, so that the complex influence relationships between nodes cannot be effectively mined. Inspired by universal gravitation, we define node gravitation by node information entropy and spatial distance, and design a novel graph attention network model based on node gravitation, which can incorporate the topological structure and node

features in a more physically meaningful way, and promote the capability of the model to extract the influence relationships between nodes. Through substantial experiments on various real datasets, it is proven that the proposed model is superior to the existing popular GNNs in node classification performance.

REFERENCES

[1] G. Corso, H. Stark, S. Jegelka, T. Jaakkola, and R. Barzilay, "Graph neural networks," *Nature Rev. Methods Primers*, vol. 4, no. 1, p. 17, 2024.

[2] H. Wang et al., "MCNE: An end-to-end framework for learning multiple conditional network representations of social network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1064–1072.

[3] T. Dai, J. Zhao, D. Li, S. Tian, X. Zhao, and S. Pan, "Heterogeneous deep graph convolutional network with citation relational bert for COVID-19 inline citation recommendation," *Expert Syst. Appl.*, vol. 213, 2023, Art. no. 118841.

[4] S. Liu et al., "Enhancing drug-drug interaction prediction using deep attention neural networks," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 20, no. 2, pp. 976–985, Mar./Apr. 2023.

[5] L. Peng et al., "Reverse graph learning for graph neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4530–4541, Apr. 2024.

[6] R. Levie, "A graphon-signal analysis of graph neural networks," in *Proc. 37th Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA, Curran Associates Inc, 2023, pp. 64482–64525.

[7] G. Li, M. Muller, A. Thabet, and B. Ghanem, "Deepgcns: Can GCNs go as deep as CNNs?," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9267–9276.

[8] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. 2nd Int. Conf. Learn. Representations*, Y. Bengio, and Y. LeCun, Eds., Banff, AB, Canada, 2014, pp. 1–14.

[9] H. Pei, B. Wei, K. C. -C. Chang, Y. Lei, and B. Yang, "GEOM-GCN: Geometric graph convolutional networks," in Proc. 8th Int. Conf. Learn. Representations, Addis Ababa, Ethiopia, Apr. *26–30*, OpenReview.net, 2020, pp. 10247–10258.

[10] Y. Zhou, H. Zheng, X. Huang, S. Hao, D. Li, and J. Zhao, "Graph neural networks: Taxonomy, advances, and trends," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 1, pp. 1–54, 2022.

[11] L. Wang, Y. Zheng, D. Jin, F. Li, Y. Qiao, and S. Pan, "Contrastive graph similarity networks," *ACM Trans. Web*, vol. 18, no. 2, pp. 1–20, 2024.

[12] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 701–710.

[13] L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, "Struc2vec: Learning node representations from structural identity," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 385–394.

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Representations, Toulon, France, Apr. 24-26, 2017, Conf. Track Proc.*, 2017, pp. 2713–2726.

[15] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 1725–1735.

[16] D. Bo, X. Wang, C. Shi, and H. Shen, "Beyond low-frequency information in graph convolutional networks," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, pp. 3950–3957.

[17] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in Proc. 6th Int. Conf. Learn. Representations, Vancouver, BC, Canada, Apr. *30- May 3, Conf. Track Proc.*, 2018, pp. 2920–2931.

[18] A. Salehi and H. Davulcu, "Graph attention auto-encoders," in *Proc. IEEE 32nd Int. Conf. Tools Artif. Intell.*, 2020, pp. 989–996.

[19] S. Yang, L. Zhang, and X. Zhang, "Fastgat: Simple and efficient graph attention neural network with global-aware adaptive computational node attention," in *Proc. ICASSP 2024-2024 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2024, pp. 5555–5559.

[20] Y. Chen, X. -Z. Xie, and W. Weng, "Content and structure based attention for graph node classification," *J. Intell. Fuzzy Syst.*, vol. 46, no. 4, pp. 8329–8343, 2024.

[21] J. Shen, Q. T. Ain, Y. Liu, B. Liang, X. Qiang, and Z. Kou, "Gtat: Empowering graph neural networks with cross attention," *Sci. Reports*, vol. 15, no. 1, 2025, Art. no. 4760.

[22] T. Nie, Z. Guo, K. Zhao, and Z. -M. Lu, "Using mapping entropy to identify node centrality in complex networks," *Physica A: Stat. Mechan. Appl.*, vol. 453, pp. 290–297, 2016.

[23] W. -L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C. -J. Hsieh, "Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 257–266.

[24] A. J. Larrazabal, N. Nieto, V. Peterson, D. H. Milone, and E. Ferrante, "Gender imbalance in medical imaging datasets produces biased classifiers for computer-aided diagnosis," *Proc. Nat. Acad. Sci.*, vol. 117, no. 23, pp. 12592–12594, 2020.

[25] S. Wang et al., "Robust and label efficient bi-filtering graph convolutional networks for node classification," *Knowl.-Based Syst.*, vol. 224, 2021, Art. no. 106891.

[26] M. Liu, H. Gao, and S. Ji, "Towards deeper graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 338–348.

[27] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[28] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.

[29] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in Proc. 8th Int. Conf. Learn. Representations, ICLR 2020, Addis Ababa, Ethiopia, Apr. *26–30*, 2020, pp. 2502–2518.

[30] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei, "AM-GCN: Adaptive multi-channel graph convolutional networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 1243–1253.

[31] Y. Yang et al., "Graph neural networks with soft association between topology and attribute," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, pp. 9260–9268.

[32] H. Chen et al., "Neighbor enhanced graph convolutional networks for node classification and recommendation," *Knowl.-Based Syst.*, vol. 246, 2022, Art. no. 108594.

[33] H. Peng, R. Zhang, Y. Dou, R. Yang, J. Zhang, and P. S. Yu, "Reinforced neighborhood selection guided multi-relational graph neural networks," *ACM Trans. Inf. Syst.*, vol. 40, no. 4, pp. 1–46, 2021.

[34] J. Wang, T. Lin, and G. Huang, "Graph quaternion-valued attention networks for node classification," in *Proc. 4th Int. Conf. Comput., Netw. Internet Things*, 2023, pp. 673–677.

[35] D. Kim and A. Oh, "How to find your friendly neighborhood: Graph attention design with self-supervision," in *Proc. 9th Int. Conf. Learn. Representations, ICLR 2021, Virtual Event, Austria, May 3–7*, 2021, pp. 1–25.

[36] Y. Peng, J. Xia, D. Liu, M. Liu, L. Xiao, and B. Shi, "Unifying topological structure and self-attention mechanism for node classification in directed networks," *Sci. Reports*, vol. 15, no. 1, 2025, Art. no. 805.

[37] T. Qiao, W. Shan, and C. Zhou, "How to identify the most powerful node in complex networks? A novel entropy centrality approach," *Entropy*, vol. 19, no. 11, 2017, Art. no. 614.

[38] W. Jiang, Y. Yang, T. Luo, W. Zhou, E. Li, and X. Zhang, "Comprehensive influence evaluation algorithm of complex network nodes based on global-local attributes," *Chin. J. Internet Things*, vol. 6, no. 3, pp. 133–145, 2022.

[39] M. Liao, Y. Liang, and R. Wang, "The gravitation-based algorithm for community detecting in large-scale social networks," in *Proc. 2020 3rd Int. Conf. Algorithms, Comput. Artif. Intell.*, 2020, pp. 1–6.

[40] S. Zhu, J. Zhan, and X. Li, "Identifying influential nodes in complex networks using a gravity model based on the h-index method," *Sci. Reports*, vol. 13, no. 1, 2023, Art. no. 16404.

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

[42] A. Bojchevski and S. Günnemann, "Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking," in Proc. 6th Int. Conf. Learn. Representations, ICLR 2018, Vancouver, BC, Canada, Apr. *30- May 3, Conf. Track Proc.*, 2018, pp. 2286–2298.

[43] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" in Proc. 10th Int. Conf. Learn. Representations, ICLR 2022, Virtual Event, Apr. *25–29*, 2022, pp. 1–26.

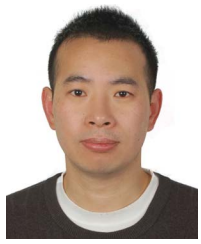
[44] T. He, Y. S. Ong, and L Bai, "Learning conjoint attentions for graph neural nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 2641–2653.

[45] W. Jin, T. Derr, Y. Wang, Y. Ma, Z. Liu, and J. Tang, "Node similarity preserving graph convolutional networks," in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, 2021, pp. 148–156.

Shujuan Wei received the B.Sc. degree in mathematics in 2023 from Tianjin University, Tianjin, China, where she is currently working toward the master's degree in mathematics with the Center for Applied Mathematics. Her research interests include Internet of Things and graph neural networks.

Huijun Tang received the B.Sc. degree from Jinan University, Guangzhou, China, in 2016, and the M.S. and Ph.D. degrees from Tianjin University, Tianjin, China, in 2018 and 2022, respectively. She is currently a Lecturer with the School of Cyberspace, Hangzhou Dianzi University. Her research interests include Internet of Things, mobile edge computing, deep learning, and complex networks.

Pengfei Jiao (Member, IEEE) received the Ph.D. degree in computer science from Tianjin University, Tianjin, China, in 2018. From 2018 to 2021, he was a Lecturer with the Center of Biosafety Research and Strategy, Tianjin University. He is currently a Professor with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou, China. His current research focuses on complex network analysis and its applications.



Huaming Wu (Senior Member, IEEE) received the B.E. and M.S. degrees in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 2009 and 2011, respectively, and the Ph.D. degree (with highest honor) in computer science from Freie Universität Berlin, Berlin, Germany, in 2015. He is currently a Professor with the Center for Applied Mathematics, Tianjin University, Tianjin, China. His research interests include mobile cloud computing, edge computing, Internet of Things, deep learning, complex networks, and DNA storage.