mathematics

MDPI

Article

MGATs: Motif-Based Graph Attention Networks

Jinfang Sheng, Yufeng Zhang, Bin Wang * and Yaoxing Chang

School of Computer Science and Engineering, Central South University, Changsha 410083, China; jfsheng@csu.edu.cn (J.S.); zhangyufeng@csu.edu.cn (Y.Z.); 214712161@csu.edu.cn (Y.C.)
* Correspondence: wb_csut@csu.edu.cn

Abstract: In recent years, graph convolutional neural networks (GCNs) have become a popular research topic due to their outstanding performance in various complex network data mining tasks. However, current research on graph neural networks lacks understanding of the high-order structural features of networks, focusing mostly on node features and first-order neighbor features. This article proposes two new models, MGAT and MGATv2, by introducing high-order structure motifs that frequently appear in networks and combining them with graph attention mechanisms. By introducing a mixed information matrix based on motifs, the generation process of graph attention coefficients is improved, allowing the model to capture higher-order structural features. Compared with the latest research on various graph neural networks, both MGAT and MGATv2 achieve good results in node classification tasks. Furthermore, through various experimental studies on real datasets, we demonstrate that the introduction of network structural motifs can effectively enhance the expressive power of graph neural networks, indicating that both high-order structural features and attribute features are important components of network feature learning.

Keywords: motifs; graph attention network; complex network; graph neural network; node classification

MSC: 68R10

Citation: Sheng, J.; Zhang, Y.; Wang, B.; Chang, Y. MGATs: Motif-Based Graph Attention Networks. *Mathematics* **2024**, *12*, 293. https://doi.org/10.3390/math12020293

Academic Editor: Jianping Gou

Received: 4 December 2023
Revised: 4 January 2024
Accepted: 12 January 2024
Published: 16 January 2024

1. Introduction

Graph neural networks (GNNs) have gained increasing popularity in recent years and continue to be highly sought after [1]. GNNs provide a universal and effective framework for data mining in complex networks, encompassing both node attribute information and edge information. GNNs are now widely applied in various domains, including social networks [2], recommendation systems [3], protein analysis [4], community detection [5], and more. In GNNs, spatial-based convolutional graph neural networks have been favored by researchers due to their powerful flexibility.

In spatial-based convolutional graph neural network models, the iteration of each node relies on the attribute features of its sampled neighboring nodes. Current research often decouples the node iteration process into sampling and aggregation operations [6,7], and different spatial-based convolution models exhibit different characteristics in these two operations. However, before the emergence of graph attention networks (GAT) [6], spatial-based convolution methods often used max pooling or average pooling as the aggregation operation, resulting in the same weight assigned to different nodes. This often neglects the differences in the impact of neighboring nodes' features on the core node during the sampling and aggregation process. GAT identified and addressed this issue by introducing graph attention mechanisms to adaptively assign weights to neighboring nodes, achieving an excellent performance in various graph tasks and making it one of the most popular spatial-based convolutional neural network models. Subsequently, based on the dynamic attention mechanism theory, the GATv2 [8] model was proposed and achieved better results in experiments.

In the field of graph neural networks, existing research often focuses on mining the features of complex real-world networks limited to node attribute features and first-order neighbor features, with insufficient attention to higher-order structural features within the network. When these graph convolution methods only focus on the local features of the network and fail to effectively utilize the global features, the trained models may suffer from overfitting. Numerous studies have shown that considering the higher-order structural features of the graph can effectively enhance the expressive power of GNN models [9,10]. Motifs [11,12] are frequently occurring subgraph structures in networks that can reflect the higher-order structural features of the network well. Therefore, this study introduces network motif structural features into the learning process of graph neural networks to enhance their local smoothness, capture weak connections between network nodes, and enhance their expressive power. Common network motif structures, such as the one shown in Figure 1, where circles represent nodes and lines represent edges, are primarily focused on in this study, with particular emphasis on the triad motif M3.

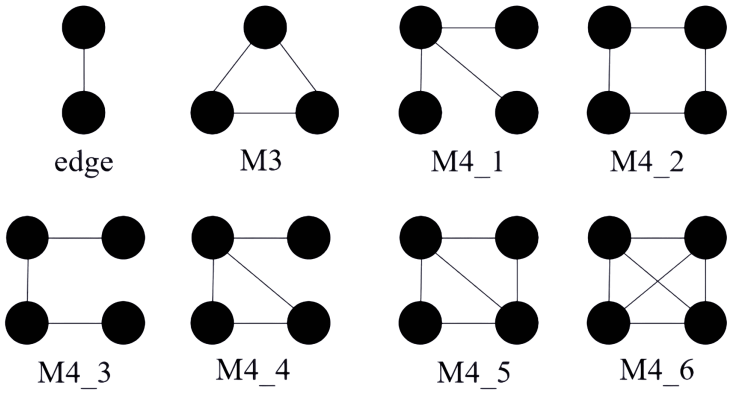


Figure 1. Partial network motif structures.

Because of the outstanding performance of the graph attention mechanism, this study combines network motif features and proposes two new motif-based graph attention models (Motif-GATs), named MGAT and MGATv2. In these two methods, this paper first obtains the graph's motif-based hybrid information matrix based on the graph's adjacency matrix and motif-based adjacency matrix. Then, this matrix is introduced into the graph attention mechanism to improve the generation process of attention coefficients, enabling the node iteration process to capture higher-order structural features of the network and obtain a novel graph attention mechanism. Finally, this paper organizes the new attention mechanism using the same multi-head attention approach as GAT to obtain a complete motif-based graph attention network model. After constructing MGAT and MGATv2, this study conducts a detailed experimental analysis on multiple real-world datasets, including node classification experiments , hyperparameter analysis experiments, and robustness experiments, which demonstrate the effectiveness of the proposed approach. Our main contributions are as follows:

- We introduce complex network motifs into the learning process of graph neural networks, allowing the models to better aggregate higher-order structural features of the graph and enhance their expressive power.
- We propose two models, MGAT and MGATv2, which achieve excellent experimental results on multiple real-world datasets by combining motif-based hybrid information matrices in the computation of attention coefficients.
- Through node classification experiments , hyperparameter experiments, and robustness experiments, we demonstrate the significant role of both higher-order structural features and attribute features in the process of graph feature learning.

2. Related Works

2.1. Graph Neural Networks

Because of the inability of traditional convolutional models to learn from graph-structured data, the early models for deep learning on graphs were based on network embedding methods, such as DeepWalk [13,14]. These methods learn low-dimensional embeddings of nodes in the Euclidean space and apply them to downstream tasks related to machine learning. However, these algorithms are unsupervised and non-end-to-end models, and they cannot incorporate node attributes, resulting in significant limitations. The first model specifically designed to handle arbitrary graph-structured data is called graph neural networks (GNNs) [15,16]. It is based on the propagation and output processes. With the remarkable success of RNNs and CNNs, it became a research focus to incorporate the advantages of other models into GNNs. One of the most successful attempts in this direction is the introduction of convolutional methods into GNNs, known as graph convolutional networks (GCNs) [17]. GCNs greatly simplify the graph convolution operation by using simple first-order filters. As graph convolution is essentially a Laplacian smoothing operation, its local smoothing can better aggregate similar information. Spatial-based graph convolution models [7,18], on the other hand, overcome the limitations of the Laplacian matrix and generalize the essence of graph convolution from the perspective of network topology as an aggregation process of neighboring node information.

2.2. Graph Attention Mechanism

Since Google achieved successful results by applying attention mechanisms in the NLP model transformer [19], attention mechanisms have become a popular research topic and have shown a remarkable performance in various fields of study. In the field of graph neural networks, there have been many attempts to integrate attention mechanisms [20], among which Graph Attention Networks (GAT) [6] can be considered the most successful one. The concise and straightforward computation of GAT makes it suitable for various graph representation learning tasks. The tremendous success of GAT has attracted scholars to further explore its potential [8,21]. In particular, the work by Brody et al. [8] pointed out the static nature of the GAT model, where the allocation of attention coefficients always assigns a lower weight to one node compared with the another, regardless of the variation in the central node. Building on this observation, they proposed a dynamic attention mechanism called GATv2, which achieved excellent results.

2.3. Motifs

Network motifs [22] are fundamental building blocks of complex networks that capture critical information about the structure and function of complex systems represented by graphs. Research in various fields has demonstrated the importance of motifs [10,23,24]. Considering the significance of motifs in the study of complex networks, algorithms for motif discovery [25] have also matured over time.

Multiple studies have shown that considering higher-order structural features is highly beneficial in various graph-based data mining tasks. Wang et al. [9] proposed the MODEL model for link prediction and analyzed the influence of different motif types on the prediction task. MODEL combines first-order neighbors and motifs with an autoencoder for learning tasks. It indirectly incorporates higher-order structural information by treating motif information as parameters in the loss function. Yu et al. [10] defined motif node degree and motif edge degree to improve traditional network representation learning using network motifs. The Motif2vec [26] model utilizes a weighted motif graph combined with random walks and Skip-Gram for heterogeneous network representation learning. John et al. [27] introduced the MCN model, which constructs a convolutional layer based on motif attention mechanisms using multiple motif matrices. It enhances the GCN model by leveraging motifs and learns the optimal motif attention through reinforcement learning, achieving excellent performance in semi-supervised node classification tasks. Rossi et al. [28] proposed the concept of higher-order network embeddings and demonstrated

that by fully considering various matrix formulations based on modality, it is possible to learn higher-quality embedding vectors.

3. Preliminaries

A directed graph can be defined as $G = (V, E)$, where $V = \{1, 2, 3..., n\}$ represents the node set and $E \in R^{|V|\times|V|}$ represents the edge set. The edge $(i, j) \in E$ represents a directed edge from node i to node j. An undirected edge can be seen as a bidirectional edge, and an undirected graph can be seen as a directed graph containing only bidirectional edges. At the same time, the initial feature of each node $i \in V$ can be represented as $h_i^0 \in R^{d0}$, and its neighbor node set is $N_i = \{j \in V|(i, j) \in E\}$. The important definitions used in this paper and their symbol representations are shown in Table 1.

Table 1. Relevant symbols and definitions used in this paper.

Symbols	Definition Description
G	Input graph G,composed of node set V and edge set E.
V	The set of all nodes in graph G.
i	A single node is represented by a lowercase letter.
N_i	The neighbor node set of node i.
E	The set of all edges in graph G.
(i, j)	Represents a directed edge from node i to node j.
A	The adjacency matrix of graph G.
h_i	The feature representation vector of node i.
e_{ij}	The attention score of neighbor node j to node i.
α_{ij}	The attention coefficient assigned to j by i after normalization of attention scores.
A_M	The motif-based adjacency matrix of graph G.
H	The motif-based mixed information matrix of graph G.
β	The proportion of adjacency matrix A in the mixed information matrix H.
W	The parameters used for learning in the model.

4. Model

4.1. Graph Attention Network

In a spatial-based convolutional graph neural network, the new feature vector of a node is obtained by continuously sampling and aggregating the features of its neighboring nodes. The input to a spatial graph convolutional layer is a node feature set $\{h_i \in R^d|i \in V\}$ and an adjacency matrix A, and the output is a new node feature set $\{h_i^{'} \in R^d|i \in V\}$. Therefore, the computation process of a spatial graph convolutional neural network layer can be formalized as follows:

$$h_i^{'} = f(h_i, AGGREGATE(\{h_j|j \in N_i\})) \quad (1)$$

The mapping function f and aggregation function $AGGREGATE$ are the most important features that differentiate one graph neural network from another. For instance, in GraphSAGE [7], the aggregation function $AGGREGATE$ typically uses average pooling, followed by concatenating the pooled output with h_i. Then, an MLP (multi-layer perceptron) is used as the mapping function f.

Before the emergence of graph attention mechanisms, graph neural networks often treated all neighboring nodes in the convolutional domain of a node equally. The aggregation function typically used average or max pooling. However, in real-world scenarios, different neighboring nodes have varying degrees of influence on the node's feature representation. Therefore, the graph attention network (GAT) introduces the concept of attention mechanisms, allowing the aggregation function to adaptively match different neighboring nodes with corresponding weights.

GAT calculates attention scores e: $R^d \times R^d \rightarrow R$ for each edge (i, j) to quantify the importance of neighbor j to node i. The higher the attention score e, the more important node j is. The computation process can be formalized as follows:

$$e_{ij} = \sigma(a^T \cdot [Wh_i || Wh_j]) \tag{2}$$

In the above formula, σ represents the non-linear activation function LeakyReLU, $a \in R^{2d'}$ and $W \in R^{d' \times d}$ are learnable parameter matrices in the model, and $||$ denotes the vector concatenation operation. After obtaining the attention scores for all of the neighboring nodes, GAT normalizes them using the softmax function in practical applications. The formulation is as follows:

$$\alpha_{ij} = softmax(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})} \tag{3}$$

In the above formula, α_{ij} represents the final attention coefficient of node j to node i, and exp() denotes the exponential function with base e.

As shown in Figure 2, in GAT, the convolutional domain is the first-order neighborhood of a node. This method assigns different attention coefficients to different neighboring node features based on their respective influence on the central node within the first-order neighborhood. Finally, the weighted sum of all neighboring node features is obtained, followed by passing through the non-linear activation function σ, resulting in the new feature h'_i of node i. The formulation is as follows:

$$h'_i = \sigma\left(\sum_{j \in N_i} \alpha_{ij} \times Wh_j\right) \tag{4}$$

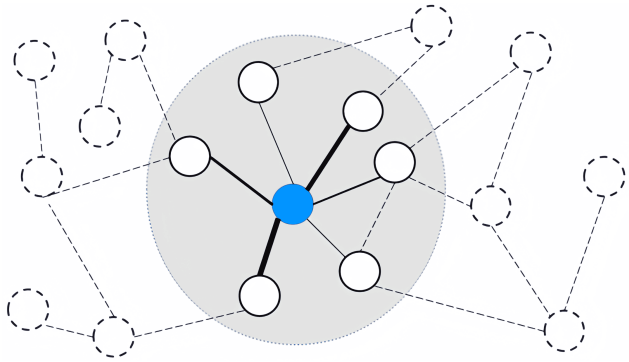


Figure 2. The influence of different neighbors in the first-order neighborhood on the central node varies.

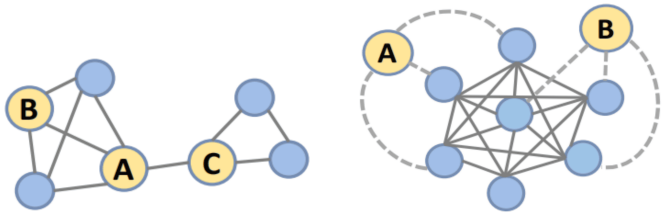
In the actual GAT model, a multi-head attention mechanism is utilized. This means that the process described above is repeated independently multiple times to compute h'_i, and after the computation is complete, the resulting h'_i vectors from the parallel and independent calculations are concatenated to form the final feature vector output. It can be observed that in GAT, the node aggregation function only focuses on the feature information of neighboring nodes and does not consider the topological structure between nodes.

4.2. Case Study

The research motivation of this paper can be explained well through the following case study.

Analysis Figure 3a: Assuming that the feature vectors of two node-pairs (A,B) and (A,C) are relatively similar from a content perspective, the content-based attention between these two groups of nodes is strong. However, from a structural analysis perspective, the

attention between (A,B) is stronger than the attention between (A,C). This is because nodes A and B are located within an interconnected community, sharing a significant proportion of common neighbors, while there are no common neighbors between nodes A and C.



(**a**) (**b**)

Figure 3. Case study analysis. (**a**) Common neighbors of two nodes; (**b**) Attachment to the same community.

Analysis Figure 3b: Nodes A and B are not direct neighbors and require three edges to connect them. Therefore, in the traditional GAT message-passing process, their messages cannot propagate to each other, meaning that their feature vectors will not be influenced by each other. However, both A and B are connected to a dense community, and node B is further connected to the center of the community. Therefore, based on the structural judgment, node A and B will directly influence each other.

In the real world, for example, by replacing the nodes in Figure 3 with products, where each edge represents two products being purchased by the same user, we can obtain the network structure of the Amazon shopping network dataset, called Amazon Photo. By performing node classification on this dataset, the system can gain a better understanding of the similarities and differences between products. However, based on the above analysis, we can draw a conclusion that relying solely on the similarity of features to calculate the attention score is insufficient. It is necessary to consider the structural details of higher-order neighbors. Therefore, this paper proposes corresponding improvement strategies to address this issue.

4.3. Proposed Model

Because of its focus on the simple and direct first-order neighborhood of nodes in the graph, GAT is prone to overfitting. In this paper, two new methods, namely MGAT and MGATv2, are proposed based on the GAT and GATv2 models, respectively. These methods introduce a hybrid information matrix based on motif structures to preserve the higher-order structural information in the graph and capture the hidden weak connections between nodes.

4.3.1. MGAT

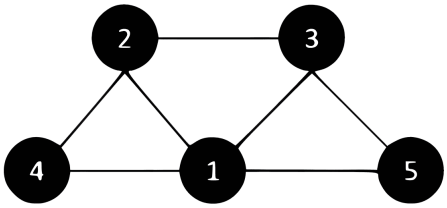
Different graph attention models have their essential differences in the computation of attention coefficients α. For instance, the graph transformer network method [13] adopts the query, key, and value mechanism from the transformer model to compute attention coefficients in graph data. In this paper, a novel approach is proposed to redesign the generation of attention coefficients α in GAT and introduce a new graph attention model.

The current convolutional graph neural network models tend to focus on low-order neighborhood structural features of graphs and ignore high-order structural features on the network. However, motifs are important high-order topological structures in the field of complex networks, and are essentially frequently occurring subgraphs, which can effectively help models capture high-order structural information of networks. This study introduces the closed triadic network structure M3 and proposes a new graph neural network, the motif-based graph attention network (MGAT). To incorporate the motif structural features while preserving the first-order neighborhood information of the

nodes, MGAT introduces the motif-based adjacency matrix A_M and the motif-based hybrid information matrix H. The formulation for calculating the A_M is as follows:

$$(A_M)_{i,j} = \begin{cases} 0, (i,j) \notin E \\ \textit{the number of motifs containing } (i,j), (i,j) \in E \end{cases} \tag{5}$$

The value at the i-th row and j-th column of the motif-based adjacency matrix A_M represents the number of closed triadic motifs in which the edge (i,j) participates. Taking the M3 motif defined in Figure 1 as an example, Figure 4a represents the original network, while Figure 4b represents the adjacency matrix A_M based on M3. From Figure 4a, it can be observed that the edge $E_{1,2}$ is included once in the M3 formed by nodes (1,2,3) and (1,2,4), thus the value of $(A_M)_{1,2}$ is 2. The construction mechanism of the motif-based adjacency matrix reveals that the more times an edge belongs to motifs, the greater its information aggregation weight. Therefore, the motif-based adjacency matrix A_M can preserve the high-order structural information of the network to a certain extent.



(**a**)

	1	2	3	4	5
1	0	2	2	1	1
2	2	0	1	1	0
3	2	1	0	0	1
4	1	1	0	0	0
5	1	0	1	0	0

(**b**)

Figure 4. Motif-based adjacency matrix. (**a**) Original network; (**b**) Adjacency matrix based on M3.

Specifically, when the motif is limited to closed triadic motifs, its calculation can be formulated as follows:

$$B = Hadamard(A, A^T) \tag{6}$$

$$A_M = Hadamard(BB, B) \tag{7}$$

In the above formula, A represents the adjacency matrix, B represents the transition matrix, and *Hadamard* denotes the Hadamard product (element-wise multiplication). In this case, the motif-based adjacency matrix A_M can represent the high-order structural features of the graph. To incorporate the high-order structural features while preserving the low-order node-edge relationships, this paper introduces the motif-based hybrid information matrix H. The calculation process of H can be formulated as follows:

$$H = \beta \cdot A + (1 - \beta) \cdot A_M \tag{8}$$

In the above formula, the hybrid information matrix H is represented as the weighted sum of the adjacency matrix A and the motif-based adjacency matrix A_M. The hyper-parameter β is used to control the proportion of high-order structural information and low-order structural information in the hybrid information matrix H. A larger value of β emphasizes the low-order structural information, while a smaller value of β emphasizes the high-order structural information. MGAT, based on the prototype of GAT, introduces motif structural information and redefines the attention calculation formula. Essentially, it calculates attention scores based on the weighted motif structural information. The formula can be expressed as follows:

$$e_{ij} = \sigma(a^T \cdot [Wh_i || H_{ij} \times Wh_j]) \tag{9}$$

where σ represents the non-linear activation function LeakyReLU.

As shown in Figure 5, the new attention coefficients in MGAT, compared with the original attention coefficients in GAT, incorporate both the attribute features of nodes and the structural features between nodes. This is achieved by introducing the motif-based hybrid information matrix H, which takes into account not only the low-order neighborhood structure, but also the high-order motif structure.

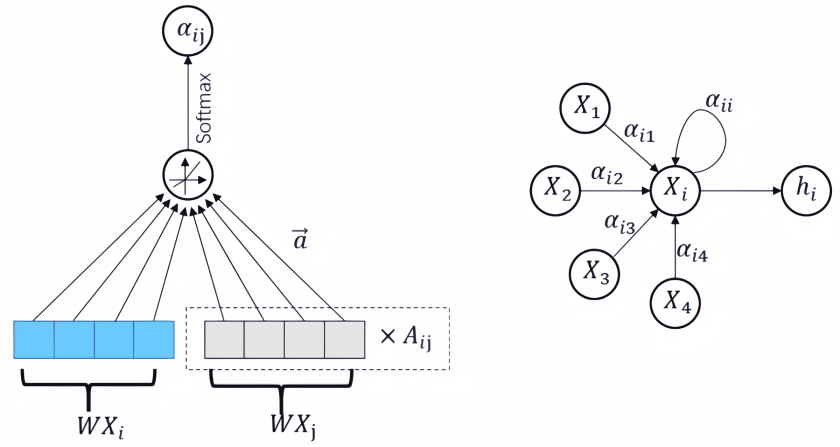


Figure 5. Motif-based graph attention mechanism.

After obtaining the new attention scores, MGAT also performs regularization on e to obtain the final attention coefficients, denoted as α. These attention coefficients are used as weights for aggregating the feature vectors of neighboring nodes in the aggregation operation. The weighted sum of the first-order neighbor features is then passed through a non-linear activation function to obtain the new node feature vectors. As shown in Figure 6, similar to GAT, MGAT also utilizes the multi-head attention mechanism.

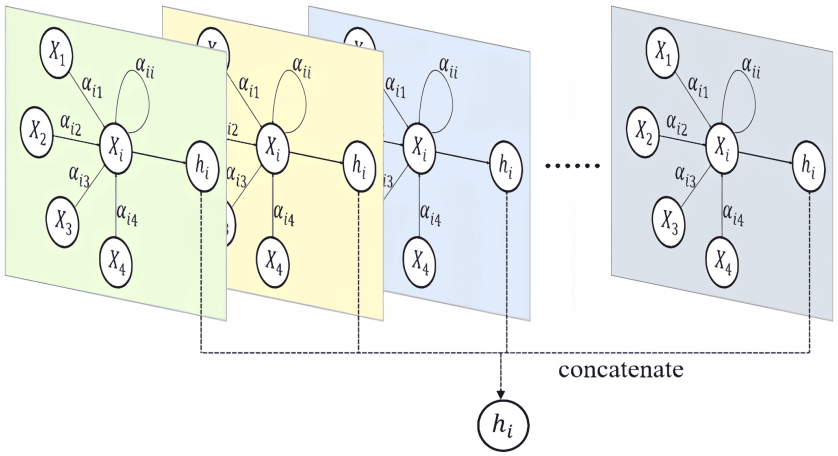


Figure 6. Multi-head attention mechanism.

A summary of MGAT is presented in Algorithm 1. Overall, the MGAT model combines node attribute features with high-order structural features, enriching the feature aggregation of the model and providing a more nuanced understanding of local attributes and global graph structures. Additionally, in MGAT, the use of the hyperparameter β balances the influence of the node attribute features and motif-based structural features, ensuring that the model does not overly prioritize one aspect. These improvements allow the MGAT model to be more adept at handling complex graph structures and enhance its ability to learn from both local and global graph features.

Algorithm 1: Summary of MGAT.

Input: Graph **G**; Node features **X**; Adjacency matrix **A**; Motif-based adjacency matrix A_M; Hyperparameter β; Number of epochs **Epoch**.
Output: Enhanced node representations **Z**.
1: Initialize MGAT with graph G, features X, matrices A, A_M, and hyperparameter β;
2: For $i = 0$ to $Epoch - 1$ do:
3: Compute mixed information matrix H using A, A_M, and β [Equation (8)];
4: # H combines motif and adjacency information for attention calculation.
5: Calculate attention scores using H [Equation (9)];
6: Normalize attention scores to obtain attention coefficients [Equation (3)];
7: Update node representations [Equation (4)];
8: Optimize model parameters (e.g., using Negative Log Likelihood Loss);
9: End for
10: Return enhanced node representations Z.

4.3.2. MGATv2

In further research on GAT, the work by Shaked Brody et al. [8] suggests that the attention mechanism in GAT is a static attention mechanism. The static nature of the attention mechanism in GAT refers to the phenomenon where different core nodes have consistent attention distribution when aggregating neighbor node features.

As shown in Figure 7, the static nature of the attention mechanism in GAT can be observed as follows: if for a specific query node q_1, the attention coefficients assigned to key nodes k_1 and k_2 are $\alpha_{11} > \alpha_{12}$, respectively, then for any query node q_1, it holds that $\alpha_{i1} > \alpha_{i2}$. This phenomenon is visually represented in the attention coefficient line graph, where, for a given sequence of key nodes, the trend of attention coefficient changes remains the same, regardless of the variation in query nodes. In contrast, for the dynamic attention mechanism in GATv2, the attention coefficients assigned by a query node to any key node are independent of any other query node in the graph. The attention coefficient line graph provides visual proof of this.

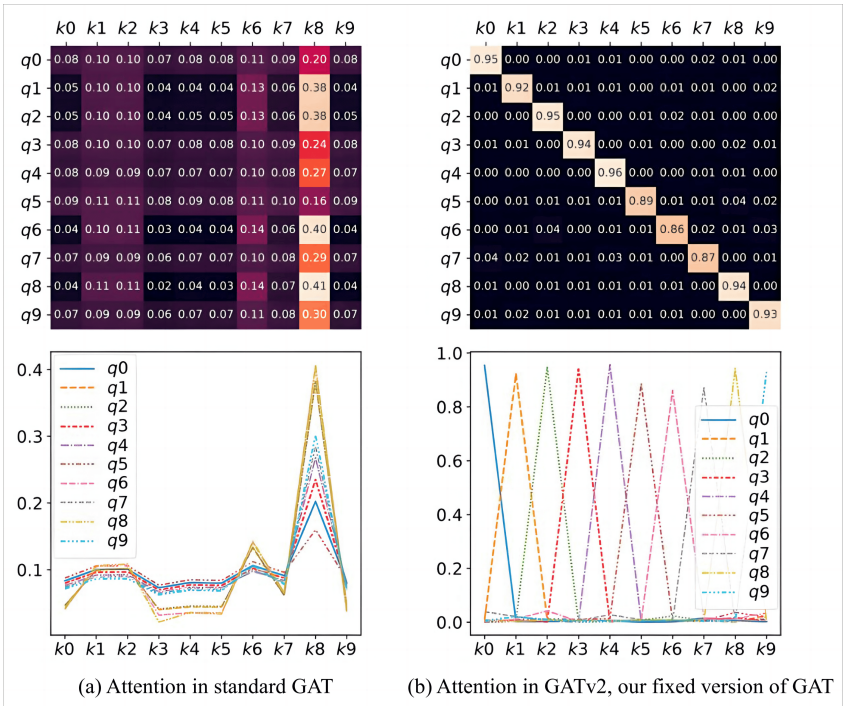


Figure 7. Comparison of static attention in GAT and dynamic attention in GATv2.

After discovering this phenomenon, the work of GATv2 proved and addressed this issue. In contrast with the static attention mechanism, it proposed a dynamic attention mechanism, where the calculation formula for attention scores e is as follows:

$$e_{ij} = a^T \cdot LeakyReLU(W \cdot [h_i||h_j]) \tag{10}$$

where the variables have the same meanings as the attention coefficient calculation formula in GAT, and the remaining steps are consistent with GAT. GATv2 demonstrated the dynamic nature of this attention calculation approach.

As shown in Figure 8, following the same consideration as MGAT, MGATv2 introduces the motif-based hybrid matrix H to enhance the expressive power of high-order structural features in the graph attention mechanism. The calculation method for attention scores e can be formulated as follows:

$$e_{ij} = a^T \cdot LeakyReLU(W \cdot [h_i||H_{ij} \times h_j]) \tag{11}$$

Similarly, MGATv2 is also a dynamic attention mechanism.

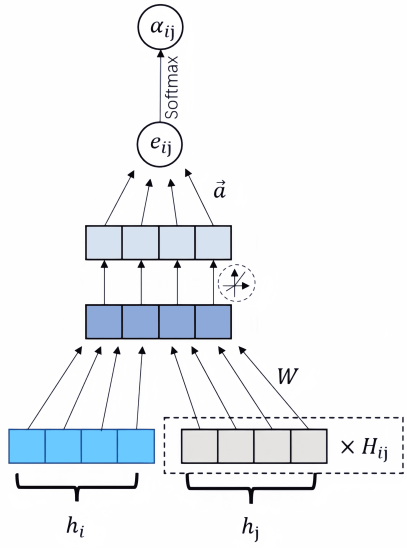


Figure 8. Motif-based GATv2.

5. Experiments

5.1. Datasets

The experiments in this paper utilized six real-world citation network datasets. Table 2 presents the statistical information of the six datasets (https://github.com/wcyszd/MGATs/tree/main/datasets, accessed on 10 January 2024). All three datasets are constructed in the same manner, where each individual article is treated as a node in the network, and the citations between articles form the edges. The original features of the nodes are represented by the bag-of-words vectors of the articles, while the research fields of the articles serve as the node categories.

Table 2. Statistical information of the datasets.

Dataset	#Classes	#Features	#Edges	#Nodes	#Training	#Validation	#Test
Cora	7	1433	5429	2708	140	500	1000
Citeseer	6	3703	4732	3327	120	500	1000
Pubmed	3	500	44,338	19,717	60	500	1000
Coauthor CS	15	6805	81,894	18,333	300	500	17,583
ogbn-arxiv	40	128	1,166,243	169,343	90,941	29,799	48,603
Amazon Photo	8	745	119,043	7487	160	240	7087

The medium-sized citation dataset, Coauthor CS, consists of an academic network that captures co-authorship relationships. In the graph, nodes represent authors, and edges represent co-authorship relationships. The node features are bag-of-words representations of paper keywords. The large-scale citation network, ogbn-arxiv, represents the citation relationships among computer science papers from the arXiv repository. In the graph, nodes represent papers, and edges represent citation relationships. The node features are embeddings of paper titles and abstracts. In addition to these datasets, there is also the Amazon Photo dataset, which represents a shopping network. In this dataset, nodes represent products and edges represent products that have been purchased together. The node features are product reviews encoded as bag-of-words, and the labels are predefined product categories.

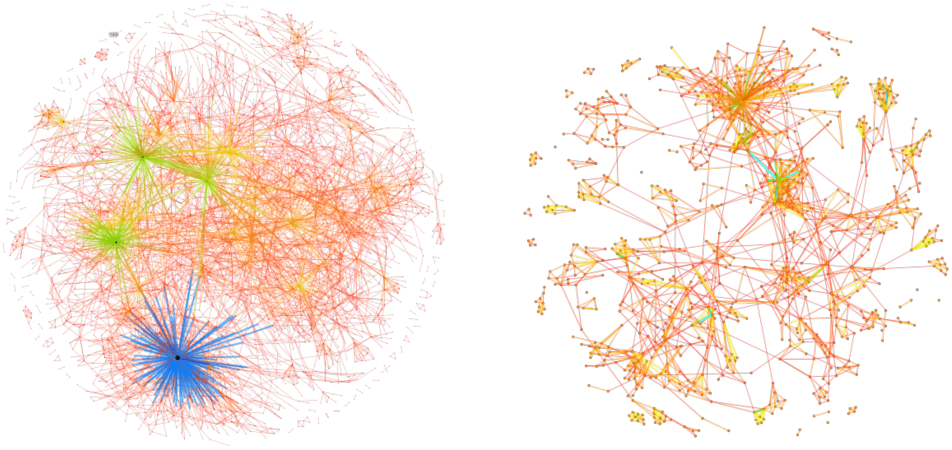
In this study, to enhance the expressive power of graph attention models for high-order structural features, the most frequent high-order structural motif, namely the three-node motif, was introduced. Therefore, this study also conducted a statistical analysis of the three-node motif structures in the six datasets, as shown in Table 3. The column "Heterogeneity Rate" refers to the ratio of motifs with incomplete label consistency among internal nodes to the total number of motifs.

Table 3. Statistical information of data motifs.

Dataset	Trimer Motifs	Heterogeneous Motifs	Motif Heterogeneity Rate
Cora	1630	283	17.35%
Citeseer	1542	313	20.32%
Pubmed	12,566	2949	23.47%
Coauthor CS	86,519	17,434	20.15%
ogbn-arxiv	4,191,164	995,731	23.76%
Amazon Photo	729,125	162,710	22.32%

In the Cora citation network, there are a total of 1630 three-node motif structures. Among them, there are 283 motifs where the three internal nodes have inconsistent labels, resulting in a motif heterogeneity rate of 17.35%. This also applies to other datasets. In terms of community divisions based on labels, these heterogeneous motifs span multiple communities and can serve as intermediaries between nodes with weaker connections. Therefore, the introduction of network motifs not only preserves high-order structural features, but also allows for better aggregation of node attribute features from nodes with weaker connections to the target node.

To visually demonstrate the importance of motif structures in the graph's topology, this study visualized the Cora dataset, as shown in Figure 9. Figure 9a and Figure 9b display the original edge information and the M3 motif information of the Cora dataset, respectively. From the visualization, it can be observed that complex network motifs play a role similar to the skeleton in the construction of the network topology. In addition to the simple edge structures between low-order nodes, the high-order topological structures represented by motifs contain rich information. They play an indispensable role in representing the global characteristic information of the entire network.



(**a**) (**b**)

Figure 9. Visualization of the Cora dataset. (**a**) Original network structure in Cora dataset; (**b**) M3 motif structure in Cora dataset.

5.2. Baseline Methods

To validate the effectiveness of the proposed model, node classification experiments were conducted on the aforementioned datasets in this study. The comparative algorithms used in the experiments included DeepWalk [29], MLP, ICA [30], LP [31], MoNet [32], GCN [17], MCN [26], GraphSAGE [7], GAT [6], and GATv2 [8], resulting in a total of 10 methods. Here is a brief introduction to these methods:

- DeepWalk: A model based on random walks that learns node embedding representations by combining the SkipGram approach from NLP methods. It is often used in semi-supervised learning tasks.
- MLP: Multi-Layer Perceptron, a fully connected neural network architecture that learns and trains node features. It is commonly used in semi-supervised tasks.
- ICA: Independent Component Analysis, a semi-supervised model that learns relationships between network nodes based on structured logistic regression.
- LP: Label Propagation, a semi-supervised model that learns pairwise node features based on Gaussian random fields and performs node classification in weighted graphs.
- MoNet: A deepened version of convolutional neural networks (CNNs) that can be applied to mining data in graph-structured data.
- GCN: Graph Convolutional Network, a spectral-based graph neural network model that performs graph convolution computations using graph signal processing filters. It can aggregate node features from first-order neighboring nodes and has been successfully applied to semi-supervised node classification tasks.
- MCN: Introduces motif attention and self-attention in graph convolution and uses reinforcement learning to obtain optimal motifs in the model.
- GraphSAGE: A spatial-based graph convolutional network model that splits spatial convolution calculations into sampling and aggregation operations, making it suitable for large-scale networks.
- GAT: Graph Attention Network, a spatial-based graph convolutional network that introduces attention mechanisms into graph convolution computations, allowing adaptive assignment of aggregation weights to different neighboring nodes.
- GATv2: A variant of GAT that proposes a dynamic attention mechanism by changing the attention coefficient calculation formula, enhancing the expressive power of the model.

All of the experiments were conducted on a Linux server equipped with 2 Intel(R) Xeon(R) Gold 6230R CPUs @ 2.10 GHz and 4 NVIDIA GeForce RTX 3090 GPUs. For the comparative algorithms, this study used the experimental parameters specified in the original works. For GCN, the learning rate (lr) was set to 0.01, the model had two

layers, and the hidden layer dimension was set to 16. For GraphSAGE, the learning rate (lr) was set to 0.01, the model had two layers, and the hidden layer dimension was 128 for the Cora and Citeseer datasets, while it was 256 for the Pubmed dataset. The batch size was set to 16. Both GCN and GraphSAGE had an L2 loss of 0.0005 during training. For ease of comparison, the training parameters for GAT and MGAT were set to be the same. The network models had two layers, the learning rate (lr) was 0.005, the number of attention heads was 8, the hidden layer dimension was 8, and the L2 loss was set to 0.0005. In MGAT, β was set to 0.7. Similarly, the training parameters for GATv2 and MGATv2 were also set to be the same. The network models had two layers, the learning rate (lr) was 0.005, the number of attention heads was 4, the hidden layer dimension was 8, and the L2 loss was set to 0.0005. In MGATv2, β was set to 0.5. For the ogbn-arxiv dataset, we employed a mini-batch strategy due to memory limitations. Using a full-batch strategy would result in insufficient memory.

5.3. Node Classification Experiment

For the node classification experiment, this study used the Accuracy (ACC) indicator to measure the experimental effect, and its calculation method is formulated as follows:

$$Accuracy = \frac{\sum_{j=1}^{n} \delta(predict(j), label(j))}{n} \tag{12}$$

$$\delta(x, y) = \begin{cases} 1, & if\ x = y \\ 0, & else \end{cases} \tag{13}$$

where predict(j) represents the predicted class of node j, and label(j) represents the true class of the node. The larger the ACC value, the better the prediction performance of the model.

The results of the classification experiments are shown in Table 4, where the numbers in bold indicate the best performance, and the numbers underlined represent the second-best performance. On the Cora dataset, GAT and GATv2 achieved classification accuracies of 83.0% and 84.0%, ranking fifth and third in terms of performance, respectively. After incorporating motif information, MGAT and MGATv2 achieved the best classification accuracy of 84.5%, sharing the top rank. On the Citeseer dataset, GAT and GATv2 achieved classification accuracies of 70.9% and 69.1%, ranking fourth and sixth, respectively. After incorporating motif information, MGAT achieved the second-best accuracy of 72.2%, while MGATv2 achieved a classification accuracy of 71.1%, ranking second and third, respectively. On the Pubmed dataset, GAT and GATv2 achieved classification accuracies of 79.2% and 80.5%, ranking fifth and third, respectively. After incorporating motif information, MGAT achieved the second-best accuracy of 82.0%, while MGATv2 achieved the best classification accuracy of 82.6%, ranking second and first, respectively.

Table 4. Node classification experiment results on the three small datasets.

Baseline Methods	Cora	Citeseer	Pubmed
DeepWalk	67.2	43.2	65.3
MLP	55.1	46.5	71.4
ICA	75.1	69.1	73.9
LP	68.0	45.3	63.0
GCN	81.5	70.3	79.0
MCN	83.5	73.3	79.3
GraphSAGE	78.9	59.0	75.0
GAT	83.0	70.9	79.2
MGAT	84.5	72.2	82.0
GATv2	84.0	69.1	80.5
MGATv2	84.5	71.1	82.6

The experimental results demonstrate that both MGAT and MGATv2, proposed in this study, outperformed the original baseline models, indicating that incorporating high-order structural motif information can improve the performance of graph neural networks. MGAT showed an accuracy improvement of 1.3% to 2.8% compared with GAT across the three datasets, while MGATv2 exhibited an accuracy improvement of 0.5% to 2.1% compared to GATv2. Compared with MCN, which also incorporates motif structural features, the performance of MGAT and MGATv2 varied across the datasets. On the Cora dataset, MGAT and MGATv2 achieved a 1.0% improvement in classification accuracy compared with MCN. On the Citeseer dataset, MGAT and MGATv2 showed a decrease in classification accuracy of 1.0% to 2.0%, while on the Pubmed dataset, an improvement of 2.7% to 3.1% was observed. This suggests that the presence of three motif structures enhanced the graph attention operations on the Cora and Pubmed datasets, and the richer the motif structures in the dataset, the better the feature extraction performance of MGAT and MGATv2.

To demonstrate the impact of MGATs on different types of networks, we also conducted node classification experiments on other networks. The results in Table 5 show that both the MGAT and MGATv2 models outperformed their baseline models on the Coauthor CS, ogbn-arxiv, and Amazon Photo datasets. On the Coauthor CS dataset, MGAT and MGATv2 achieved accuracies of 86.7% and 85.5%, respectively. Notably, on the large-scale citation network ogbn-arxiv, MGATv2 achieved the highest accuracy of 74.3%, demonstrating its effectiveness in handling large datasets. Additionally, MGATv2 achieved a high accuracy of 93.7% on the Amazon Photo dataset, outperforming the other methods.

Table 5. Classification accuracy (%) and execution time (minutes) of all methods on the three datasets.

Methods	Coauthor CS		ogbn-arxiv		Amazon Photo	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
Deep Walk	85.3	4.4	63.6	10.2	89.4	2.6
GCN	84.5	0.4	70.4	0.2	91.6	0.2
GraphSAGE	85.1	8.2	63.6	18.8	91.0	3.6
GAT	85.7	3.0	70.6	5.0	91.8	1.6
MGAT	86.7	3.3	73.5	5.7	93.4	1.8
GATv2	84.3	2.1	71.8	3.9	92.2	1.1
MGATv2	85.5	2.3	74.3	4.5	93.7	1.4

In terms of computational resources, although the introduction of MGAT and MGATv2 resulted in a slight increase in resource usage, this increase was not significant. Even when handling large datasets like ogbn-arxiv, the increase in computational resources was relatively limited. This indicates that while MGAT and MGATv2 require more computational resources when dealing with complex and large datasets, this increase is reasonable and manageable, as it improves the model's performance without significantly increasing computational costs.

To investigate how motif structural information enhances the expressive power of graph attention mechanisms, this study compared the experimental results on six datasets. As shown in Figure 10, both MGAT and MGATv2 exhibited the highest improvement on the ogbn-arxiv dataset, followed by the Pubmed dataset. Furthermore, in the dataset statistical analysis, ogbn-arxiv and Pubmed also had the highest motif label inconsistency rates among the six datasets. This indicates that the motifs involving multiple communities in the network better reflect the high-order structural features of complex networks and effectively enhance the expressive power of graph attention networks.

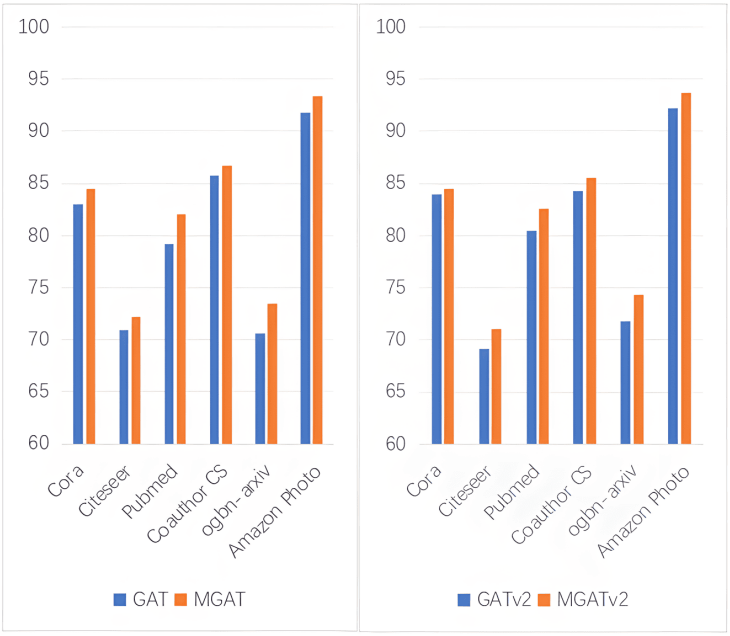


Figure 10. Comparison of experimental results between MGATs and their respective baseline algorithms.

It is commonly believed that the number of common neighbors between two vertices can be used to measure their similarity. The more common neighbors they have, the more similar the nodes are. However, in real-world networks, there are cases where some vertices exhibit weak connections, meaning they have a high similarity but share few or no common neighbors. This phenomenon is visually represented by the fact that the two vertices belong to different communities. When only considering the first-order neighborhoods of nodes, it is difficult to discover the potential weak connections between two vertices. When the labels of nodes within a motif are inconsistent, the motif spans multiple different communities. In this case, such a heterogeneous motif can be seen as a bridge between communities, capturing the weak connections between nodes and effectively reducing the overfitting of the method. As a result, the model has a higher expressive power.

5.4. Hyperparametric Experiment

This paper investigates the impact of the hyperparameter β on the expressive power of MGAT in node classification experiments on the Cora dataset. The hyperparameter β controls the ratio of motif information and first-order neighbor information in the mixed information matrix. A larger value of β provides more weight to the first-order neighbor information in the mixed information matrix. When $\beta = 1$, the mixed information matrix becomes the adjacency matrix. Conversely, a smaller value of β provides more weight to the motif information in the mixed information matrix. When $\beta = 0$, the mixed information matrix becomes the motif-based adjacency matrix. The range of β starts from 0 and ends at 1.0, with an increment of 0.1, intuitively demonstrating the influence of the ratio of motif information to low-order information in the mixed information matrix on the effectiveness of graph convolution.

The experimental results, as shown in Figure 11, demonstrate that when β is 0.7 and 0.8, representing a ratio of 3:7 and 2:8 between motif information and first-order neighbor information, respectively, the node classification task achieves the best performance of 84.5%. Conversely, the worst performance is obtained when only the adjacency matrix or the motif-based adjacency matrix is used. Similarly, for MGATv2, with β being 0.4 and 0.5, corresponding to a ratio of 4:6 and 5:5 between motif information and first-order neighbor information, respectively, the node classification task achieves the best performance of 84.5%. Once again, the worst performance is observed when only the adjacency matrix or the motif-based adjacency matrix is used. These findings indicate that both the high-

order structural motif features and the simple edge features of first-order neighbors have a significant impact on graph convolution computations.

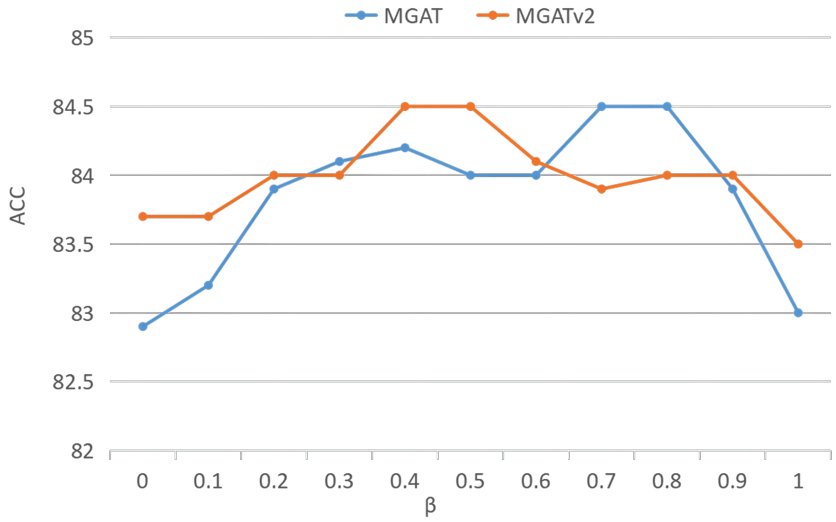


Figure 11. The relative relationship between hyperparameters and node classification accuracy.

5.5. Robustness to Noise

This study examines the robustness of the MGATs model to noise. GIven an input graph $G = (V, E)$, a noise ratio of $0 \leqslant p \leqslant 1$ is set, and $|E| \times p$ non-existent edges E_0 are randomly sampled from $V \times V \backslash E$. Finally, the model is trained on the noisy graph $G_0 = (V, E \cup E_0)$.

Figure 12 demonstrates the performance variation of the MGATs model compared with the GAT and GATv2 models when introducing different levels of noise in the graph data on the Pubmed dataset. The experimental results show that as the noise increases, the performance of MGATs decreases significantly less compared with the GAT and GATv2 models. This improved robustness can be attributed to the utilization of motifs, which enhances the model's stability against local disturbances by capturing higher-order relationships that are less sensitive to local perturbations. Additionally, the motif-based feature aggregation approach in MGATs allows for a more diversified handling of features compared with the GAT and GATv2 models. This diversity helps dilute the impact of noise.

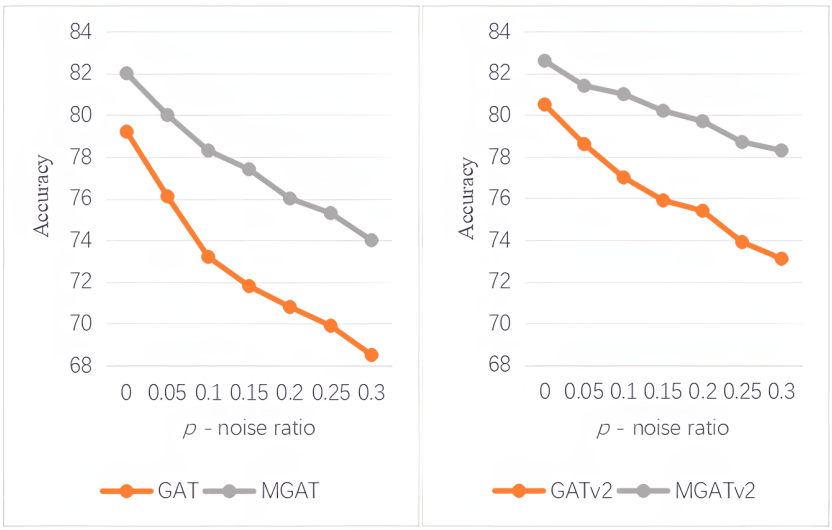


Figure 12. Experimental results of MGAT's robustness to noise.

6. Conclusions

The main innovation and contribution of this paper lie in the discovery that existing graph attention models lack attention to the high-order structure of networks. As a result, the MGAT and MGATv2 models are proposed, which capture the high-order structural features of networks by introducing motif information, thereby improving the efficiency of graph attention model convolutional aggregation. In the node classification experiments on multiple datasets, both MGAT and MGATv2 show significant improvements compared with their baseline models. Through hyperparameter experiments and robustness analysis, this paper demonstrates the indispensable role of both low-order and high-order structural features in the process of graph feature learning. Furthermore, it is shown that the introduction of high-order structural features effectively enhances the robustness of the model. These experiments validate the effectiveness of this work, showing that motif structures are important in networks, and that high-order structural features can effectively enhance the expressive power of graph neural networks. However, it is true that the MGATs model does require additional computational resources when dealing with complex network structures. This aspect will be a key focus of our attention in future optimizations. Additionally, this paper finds that on the Citeseer dataset, MGAT and MGATv2 perform slightly worse than MCN, which also incorporates motif information. This may be attributed to MGAT and MGATv2 only introducing one type of motif, namely the three-motif. In the future, we may consider incorporating more types of motifs to further improve the model performance.

Author Contributions: Conceptualization, J.S. and B.W.; methodology, Y.Z.; validation, Y.Z. and Y.C.; formal analysis, Y.Z. and J.S.; writing—original draft preparation, Y.C.; writing—review and editing, J.S., Y.Z. and B.W.; visualization, Y.Z.; supervision, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Key Research and Development Program of Hunan Province (Grant No. 2023SK2038).

Data Availability Statement: Data sharing are not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wu, L.; Cui, P.; Pei, J.; Zhao, L.; Guo, X. Graph neural networks: Foundation, frontiers and applications. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 4840–4841.
2. Li, X.; Sun, L.; Ling, M.; Peng, Y. A survey of graph neural network based recommendation in social networks. *Neurocomputing* **2023**, *549*, 126441. [CrossRef]
3. Gao, C.; Wang, X.; He, X.; Li, Y. Graph neural networks for recommender system. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Tempe, AZ, USA, 21–25 February 2022; pp. 1623–1625.
4. Jha, K.; Saha, S.; Singh, H. Prediction of protein–protein interaction using graph neural networks. *Sci. Rep.* **2022**, *12*, 8360. [CrossRef]
5. Souravlas, S.; Anastasiadou, S.; Katsavounis, S. A survey on the recent advances of deep community detection. *Appl. Sci.* **2021**, *11*, 7179. [CrossRef]
6. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *Stat* **2017**, *1050*, 10-48550.
7. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1024–1034.
8. Brody, S.; Alon, U.; Yahav, E. How Attentive Are Graph Attention Networks? *arXiv* **2021**, arXiv:2105.14491.
9. Wang, L.; Ren, J.; Xu, B.; Li, J.; Luo, W.; Xia, F. Model: Motif-based deep feature learning for link prediction. *IEEE Trans. Comput. Soc. Syst.* **2020**, *7*, 503–516. [CrossRef]
10. Yu, S.; Xia, F.; Xu, J.; Chen, Z.; Lee, I. Offer: A motif dimensional framework for network representation learning. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Online, 19–23 October 2020; pp. 3349–3352.
11. Milo, R.; Shen-Orr, S.; Itzkovitz, S.; Kashtan, N.; Chklovskii, D.; Alon, U. Network motifs: Simple building blocks of complex networks. *Science* **2002**, *298*, 824–827. [CrossRef] [PubMed]
12. Jain, D.; Patgiri, R. Network motifs: A survey. In Proceedings of the Advances in Computing and Data Sciences: Third International Conference, ICACDS 2019, Ghaziabad, India, 12–13 April 2019; Revised Selected Papers, Part II 3; Springer: Berlin/Heidelberg, Germany, 2019; pp. 80–91.

13. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1067–1077.
14. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
15. Gori, M.; Monfardini, G.; Scarselli, F. A new model for learning in graph domains. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005; Volume 2, pp. 729–734.
16. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2008**, *20*, 61–80. [CrossRef] [PubMed]
17. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907.
18. Chen, J.; Ma, T.; Xiao, C. Fastgcn: Fast Learning with Graph Convolutional Networks Via Importance Sampling. *arXiv* **2018**, arXiv:1801.10247.
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
20. Dwivedi, V.P.; Bresson, X. A Generalization of Transformer Networks to Graphs. *arXiv* **2020**, arXiv:2012.09699.
21. Rong, Y.; Bian, Y.; Xu, T.; Xie, W.; Wei, Y.; Huang, W.; Huang, J. Self-supervised graph transformer on large-scale molecular data. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12559–12571.
22. Yuan, H.; Yu, H.; Wang, J.; Li, K.; Ji, S. On explainability of graph neural networks via subgraph explorations. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 18–24 July 2021; pp. 12241–12252.
23. Bao, X.; Hu, Q.; Ji, P.; Lin, W.; Kurths, J.; Nagler, J. Impact of basic network motifs on the collective response to perturbations. *Nat. Commun.* **2022**, *13*, 5301. [CrossRef] [PubMed]
24. Lotito, Q.F.; Musciotto, F.; Montresor, A.; Battiston, F. Higher-order motif analysis in hypergraphs. *Commun. Phys.* **2022**, *5*, 79. [CrossRef]
25. Lotito, Q.F.; Musciotto, F.; Battiston, F.; Montresor, A. Exact and sampling methods for mining higher-order motifs in large hypergraphs. In *Computing*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 1–20.
26. Dareddy, M.R.; Das, M.; Yang, H. motif2vec: Motif aware node representation learning for heterogeneous networks. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 1052–1059.
27. Lee, J.B.; Rossi, R.A.; Kong, X.; Kim, S.; Koh, E.; Rao, A. Graph convolutional networks with motif-based attention. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 499–508.
28. Rossi, R.A.; Ahmed, N.K.; Koh, E.; Kim, S.; Rao, A.; Yadkori, Y.A. Hone: Higher-order network embeddings. *arXiv* **2018**, arXiv:1801.09303.
29. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
30. Atashpaz-Gargari, E.; Lucas, C. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 4661–4667.
31. Zhu, X.; Ghahramani, Z.; Lafferty, J.D. Semi-supervised learning using gaussian fields and harmonic functions. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), Washington, DC, USA, 21–24 August 2003; pp. 912–919.
32. Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; Bronstein, M.M. Geometric deep learning on graphs and manifolds using mixture model cnns. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5115–5124.