

Article

# MBHAN: Motif-Based Heterogeneous Graph Attention Network

Qian Hu <sup>1,2,\*</sup>, Weiping Lin <sup>2</sup>, Minli Tang <sup>2</sup>  and Jiatao Jiang <sup>3</sup><sup>1</sup> School of Media and Communications, Guizhou Normal University, Guiyang 550000, China<sup>2</sup> School of Informatics, Xiamen University, Xiamen 361000, China; linweiping@stu.xmu.edu.cn (W.L.); tangml@stu.xmu.edu.cn (M.T.)<sup>3</sup> School of mathematical Science, Guizhou Normal University, Guiyang 550000, China; jjt@gznu.edu.cn

\* Correspondence: huqian@gznu.edu.cn

**Abstract:** Graph neural networks are graph-based deep learning technologies that have attracted significant attention from researchers because of their powerful performance. Heterogeneous graph-based graph neural networks focus on the heterogeneity of the nodes and links in a graph. This is more effective at preserving semantic knowledge when representing data interactions in real-world graph structures. Unfortunately, most heterogeneous graph neural networks tend to transform heterogeneous graphs into homogeneous graphs when using meta-paths for representation learning. This paper therefore presents a novel motif-based hierarchical heterogeneous graph attention network algorithm, MBHAN, that addresses this problem by incorporating a hierarchical dual attention mechanism at the node-level and motif-level. Node-level attention aims to learn the importance between a node and its neighboring nodes within its corresponding motif. Motif-level attention is capable of learning the importance of different motifs in the heterogeneous graph. In view of the different vector space features of different types of nodes in heterogeneous graphs, MBHAN also aggregates the features of different types of nodes, so that they can jointly participate in downstream tasks after passing through segregated independent shallow neural networks. MBHAN's superior network representation learning capability has been validated by extensive experiments on two real-world datasets.

**Keywords:** heterogeneous graphs; graph neural networks; representation learning; motif; attention mechanism



**Citation:** Hu, Q.; Lin, W.; Tang, M.; Jiang, J. MBHAN: Motif-Based Heterogeneous Graph Attention Network. *Appl. Sci.* **2022**, *12*, 5931. <https://doi.org/10.3390/app12125931>

Academic Editors: Katia Lida Kermanidis, Phivos Mylonas and Manolis Maragoudakis

Received: 10 May 2022

Accepted: 9 June 2022

Published: 10 June 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Graph neural networks (GNNs) have attracted extensive attention in academia as a powerful way of approaching deep representation learning for graph data. They have been proven to perform especially well in network analysis [1,2]. The basic idea of a graph neural network is to undertake representation learning on the nodes themselves, according to their local neighborhood information. This involves aggregating the information of each node and its surrounding nodes through a neural network. So, in [3–5], the node features and graph structure in graphs are used to learn node embeddings. Convolutional operations can also be introduced into graph representation learning [6–9].

Alongside GNNs, a significant amount of interest has been shown to attention mechanisms [10], which encourage models to focus on the most salient parts of the data that will affect downstream tasks. Attention mechanisms have been highly effective when incorporated into deep neural network frameworks and are widely used across a range of different domains [11–15]. Graph Attention Networks (GAT) [16] assume that different neighboring nodes may play different roles for the core nodes. A self-attention mechanism can therefore be employed [17] to aggregate neighbor nodes and achieve an adaptive matching of weights that captures the different importance of different neighbors. However, GAT can only be applied to homogeneous graphs and cannot be easily migrated to heterogeneous graphs.

Heterogeneous graph-based neural network representation learning methods are a natural extension of deep learning approaches to the processing of structured graph data. The most popular approach, here, is to transform heterogeneous graphs into homogeneous graphs for representation learning through meta-paths [18]. Wang et al. [19], for instance, introduced a two-level hierarchical attention mechanism in graph neural networks where the node-level attention captures the relationship between neighboring nodes generated by a certain meta-path, while meta-path semantic attention captures the importance of the different meta-paths in the original heterogeneous graph. To make the most of the rich interaction information present in heterogeneous graphs focused on intent recommendation, Fan et al. [20] used meta-path guided neighbors to aggregate the node information and designed different aggregation functions based on different types of neighbor features. All of these methods require an expert to design and manually set the meta-paths for specific issues. There is also inevitably a loss of information during the transformation between heterogeneous and homogeneous graphs, and the selection of different meta-paths can lead to significant performance fluctuations in downstream tasks [21].

Numerous studies [22–30] have verified the superior performance of motifs as fundamental building blocks of non-random real-world graph-structured data for the purposes of graph representation learning. A motif is essentially a subgraph structure consisting of multiple nodes and links. The semantic relationships among the set of nodes that make up a motif are particularly closely related. Thus, it seems reasonable to assume that, if the representation of nodes in motif-based subgraphs with different structural patterns can be captured, it will be possible to characterize different types of nodes in heterogeneous graphs from multiple perspectives. The Multiscale Convolutional Network (MCN) [30] constructed graph convolutional neural network introduces multiple weighted motif-based adjacency matrices to capture higher-order neighborhood information. Peng et al. [31] used motifs for subgraph normalization and designed the Motif-based Attentional Graph Convolutional Neural Network (MA-GCNN) for subgraph classification tasks. However, the MCN, here, is still constrained to the pre-design of the motif structure by an expert and does not consider the motif schemas of other morphologies in the graph. The MCN is also not scalable to heterogeneous graphs. MA-GCNN similarly pays no attention to the heterogeneity of nodes in the graph and can only be applied to subgraph classification tasks.

In view of the above, and inspired by the Heterogeneous Graph Attention Network (HAN) [19], this paper proposes an end-to-end Motif-based Heterogeneous Attention Network, MBHAN, which employs a hierarchical attention mechanism that includes node-level and motif-level attention. MBHAN is able to focus on the degree of importance of both a node's neighbors and the motif subgraph where the node is located.

The MBHAN approach presented in this paper offers the following contributions:

It efficiently learns multi-perspective representations for all the nodes in a heterogeneous graph, without having to artificially establish any meta-paths (or subgraphs).

It avoids the knowledge loss in the conversion of heterogeneous graphs to homogeneous graphs and can learn the global node information in heterogeneous graphs.

It can capture the subtle effects of different node types on downstream tasks, leading to more accurate knowledge mining.

## 2. Related Work

### 2.1. Graph Neural Networks

Graph representation learning, also known as graph embedding, aims to represent nodes in a network as low-dimensional dense vectors with real values, and the results can then be stored in a vector space. The resulting vector representations can be easily and conveniently used as the input for machine learning models, which can then be applied to common social network applications, such as visualization, node classification, link prediction, and community discovery [32]. The impressive results of neural networks in solving Euclidean space data (e.g., for images [33] and text [34], etc.) have resulted in proposals to extend deep neural networks to the processing of data with graph (network)

structures [4,5]. However, applying graph neural network techniques to heterogeneous graphs and distinguishing the heterogeneity of nodes and links in the graph structure has consistently proved challenging.

The first attempt to model graph structure data with multiple correlations using GNNs involved Relational Graph Convolutional Networks (R-GCNs) [6], which maintain unique linear mapping weights for each different type of link. They also decompose relationship-specific parameters into linear combinations of several elementary matrices, so as to be able to deal with networks with a large number of relationships. To handle the structure and node properties of heterogeneous graphs, Zhang et al. [35] began by introducing a random walk with a restart strategy that can extract a fixed number of strongly correlated heterogeneous neighbors for each node and group them according to node types. They then used a type-specific Recurrent Neural Network (RNN) to encode the vertex features of each type of neighbor. The encoded representations of different types of neighbors were then aggregated using another RNN. To address the problem of heterogeneous graph GNNs still needing to artificially set meta-paths, with a consequent loss of accuracy in downstream tasks, Yun et al. [21] proposed a graph transformation network (GTN) that is capable of generating new network structures. It can identify useful interactions between unconnected nodes in the original graph, while learning valid node representations in the transformed graph in an end-to-end fashion. A similar approach that does not involve setting meta-paths is the Heterogeneous Graph Transformer (HGT) [36], which has parameters relating to different types of nodes and links, so as to be able to characterize the heterogeneous attention for each link. This enables HGT to generate specialized representations for different types of nodes and links.

Most existing graph neural networks based on heterogeneous graphs tend to use meta-paths to transform the heterogeneous graphs into homogeneous graphs for representation learning. This inevitably results in a lack of non-vertex information in the meta-paths. At present, most methods cannot perform global node representation learning.

## 2.2. Motifs

The notion of graph motifs, which are higher-order structures in a network, was first proposed by Milo et al. [27]. They are small subordinate structures consisting of multiple nodes. In real-world applications, motifs play a critical role in complex graph analysis. Benson et al. [37] used motifs to analyze higher-order clustering in the *Caenorhabditis elegans* neuronal network and the higher-order spectral network of airports in Canada and the United States. Zhou et al. [38] looked at how a star motif structure might correspond to a synthetic counterfeit personal account number in a bank's customer information network. Other studies [39,40] have shown that triangles consisting of three nodes form the basic motif structure in most real-world networks and that this plays an important role in network formation and evolution. Some motif-based representation learning methods have also been proposed. Motif2vec [25], for instance, aggregated and shuffled random walk sequences created for both a motif-based higher-order graph and an original graph. The ultimate sequences were then fed to a Skip-Gram model [41] to learn the node embedding.

In this paper, we focus on all the triangle motif schemas in the graphs we are working with and assemble all the "atomic-level" higher-order heterogeneous connectivity patterns, so as to eliminate any interference by man-made semantic assignments.

## 3. Preliminary Information

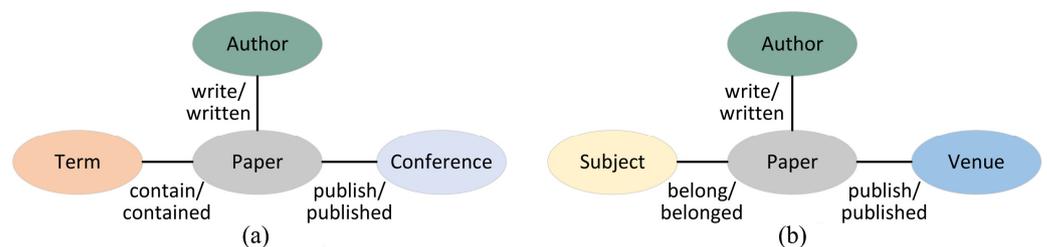
Table 1 gives the notations used in this paper and their corresponding explanations. We will then define the most relevant concepts and the problem we are seeking to address, before introducing the MBHAN algorithm.

**Table 1.** Notations and Explanations.

Notation	Explanation
$\mathcal{G}$	Heterogeneous graph
$\mathcal{V}$	Node set
$\mathcal{E}$	Link set
$v_i$	The $i$ -th node
$\mathcal{T}$	Type of node set
$\mathcal{R}$	Type of link set
$m_i$	The $i$ -th motif pattern
$\mathcal{G}_m$	The subgraph in $\mathcal{G}$ that satisfies the motif pattern, $m$
$\mathcal{N}_i^m$	The set of neighboring nodes of $v_i$ in $\mathcal{G}_m$
$h$	Node features
$e_{ij}^m$	Importance of node pair $(v_i, v_j)$ in $\mathcal{G}_m$
$a_m$	Node-level attention vector in $\mathcal{G}_m$
$a_{ij}^m$	Weight of $\mathcal{G}_m$ -based node pair $(v_i, v_j)$
$q$	Motif-level attention vector
$w_{m^t}$	Importance of $t$ -type nodes in $\mathcal{G}_m$
$\beta_{m^t}$	Attention weight of $t$ -type nodes in $\mathcal{G}_m$
$\mathcal{Z}^t$	Final representation vector of the $t$ -type nodes

**Definition 1.** *Heterogeneous graphs [42]: A heterogeneous graph is a network data structure,  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , with a node type that can be mapped as  $\varphi : \mathcal{V} \rightarrow \mathcal{T}$  and a link type that can be mapped as  $\psi : \mathcal{E} \rightarrow \mathcal{R}$ , where  $|\mathcal{T}| > 1$  represents the number of node types in the network or  $|\mathcal{R}| > 1$  represents the number of link types in the network. If  $|\mathcal{T}| = 1$  and  $|\mathcal{R}| = 1$ , this indicates that the graph,  $\mathcal{G}$ , is a homogeneous network.*

**Example 1.** *Typical examples of heterogeneous graphs are academic citation networks (see Figure 1). In Figure 1a, DBLP consists of four different types of nodes (author (A), paper (P), conference (C), and term (T)) and multiple types of links (A – P : an author writes a paper or a paper is written by an author; P – C : a paper is published in a conference or a conference publishes the paper; T – P : a paper contains a term or a term is mentioned in a paper). Figure 1b again contains multiple types of nodes, e.g., author (A), paper (P), topic (S), and publication venue (V). These are similarly connected by multiple types of links. It is worth noting that in these kinds of academic citation heterogeneous graphs, different types of nodes have different feature spaces. So, for the DBLP and ACM datasets used in this paper, only the paper (P) type nodes have initial features (bag-of-words vectors).*

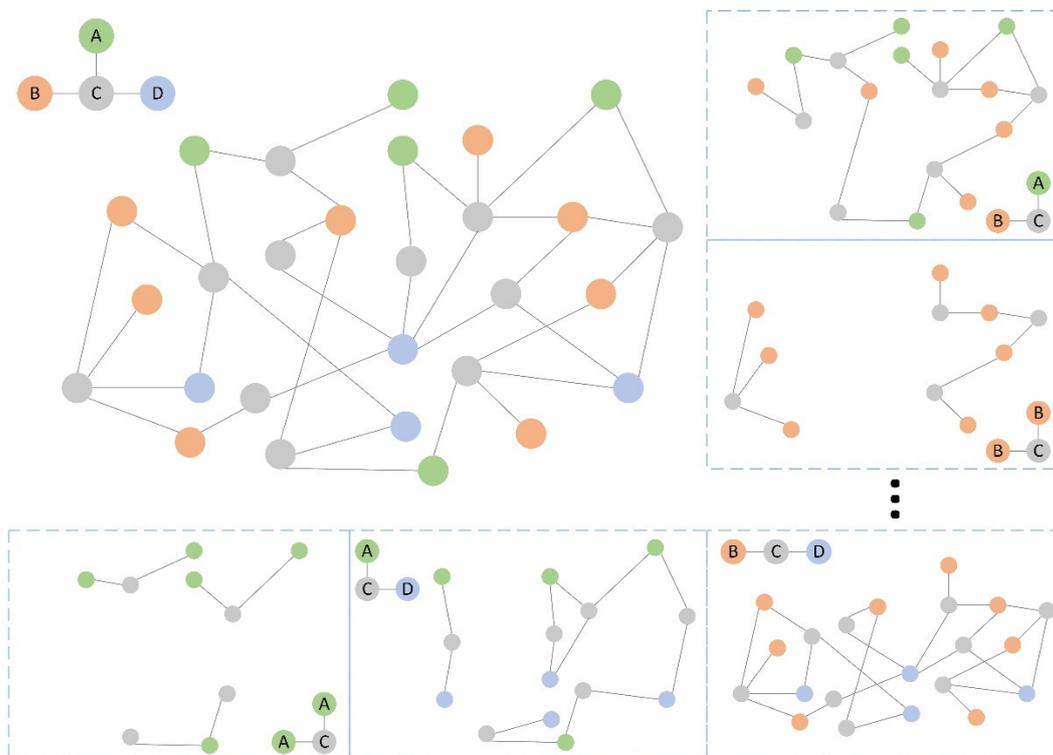


**Figure 1.** Academic citation heterogeneous graphs for (a) DBLP and (b) ACM.

**Definition 2.** *Motif subgraphs: To be able to characterize different types of nodes under different motif patterns and from multi-perspectives in the representation learning process,  $\mathcal{G}_M = (\mathcal{V}_M, \mathcal{E}_M)$  can be defined as the motif subgraph corresponding to the motif pattern,  $M$ , where  $M$  is one of the motif patterns in  $\mathcal{G}$ ,  $\mathcal{V}_M \in \mathcal{V}$ , and  $\mathcal{E}_M \in \mathcal{E}$ . The motif pattern,  $M$ , has a fixed form that can be naturally observed when the structure of the heterogeneous graph,  $\mathcal{G}$ , has been determined.*

**Example 2.** *MBHAN focuses on a motif pattern that consists of three nodes while taking into account node heterogeneity (see Figure 2). Given prior knowledge of the structural schema of the*

heterogeneous graph, the motif subgraph is a subset of the original heterogeneous graph based on different node types and compositions. Unlike the motif-based representation learning method, MBRep [24], where all the motif instances that satisfy a specific motif pattern are extracted, the node-level attention learning process of MBHAN can be performed entirely on the motif subgraph.

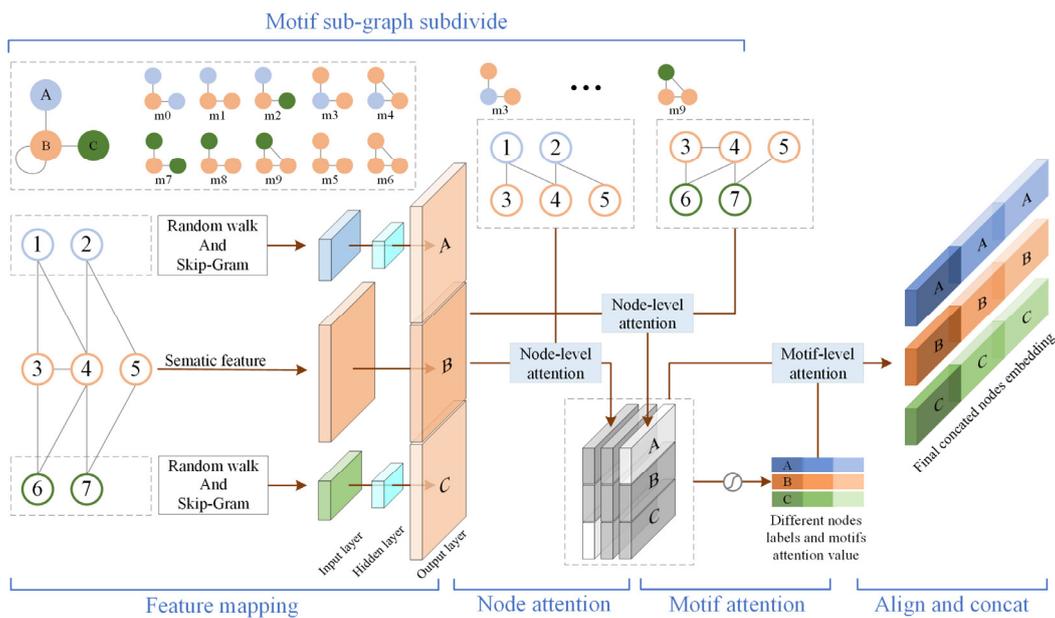


**Figure 2.** Illustration of a heterogeneous graph for four different node types and their corresponding motif subgraphs.

As noted previously, there is a tendency for heterogeneous graph-based graph neural network methods to transform heterogeneous graphs into homogeneous graphs for representation learning via meta-path relationships, making it impossible to undertake global node representation learning. To deal with this, we propose the motif-based hierarchical attention graph neural network algorithm, MBHAN, which is an end-to-end global learning model and that can excavate subtle disparities in the magnitude of attention applicable to different node levels and motif subgraphs.

#### 4. The Proposed Method

Figure 3 shows the MBHAN algorithm framework. Several aspects of the MBHAN algorithm will be presented in this section, including its node-level attention mechanism (4.1), its motif subgraph-level attention mechanism (4.2), and its nodes features mapping mechanism (4.3).



**Figure 3.** The MBHAN framework. The different colors in the heterogeneous graph represent different node types. The different colors in the hierarchical attention mechanism represent different levels of attention.

#### 4.1. Node-Level Attention Mechanism

Before aggregating different motif subgraphs for different aspects of the nodes’ representations, we first focus on the different roles played by the nodes’ neighbors in each motif subgraph. To that end, we will begin by looking at the significance features aggregated by the neighbors of each node in a specific motif subgraph. MBHAN employs a self-attention mechanism [10] that can learn the weights between different nodes. Given a motif subgraph,  $\mathcal{G}_m$ , containing a pair of nodes,  $(v_i, v_j)$ , the node-level attention,  $e_{ij}^m$ , can be defined as follows:

$$e_{ij}^m = att_{node-level}(h_i, h_j; m), \tag{1}$$

where  $e_{ij}$  denotes the importance of node  $v_j$  to node  $v_i$ ;  $att_{node-level}$  refers to the deep neural network that performs the node-level attention [16]; and, for a given motif subgraph,  $\mathcal{G}_m$ ,  $att_{node-level}$  is shared for all its node pairs. From Equation (1), it can be seen that the attention level of the node pairs  $(v_i, v_j)$  within the motif subgraph depends on their features. Unlike the approach adopted in [19], the types of node pairs are not necessarily the same in MBHAN. For example, in the “A – C – B” motif pattern subgraph in Figure 2, a node of type A is represented by aggregating the features of its first-order neighbors, which are type C nodes. The feature space transformations for different types of nodes will be presented in Section 4.3. Note also that  $v_i$  and  $v_j$  are asymmetric. So, the degree of importance of node  $v_i$  to node  $v_j$  is not necessarily the same as the degree of importance of node  $v_j$  to node  $v_i$ . This is a fundamental property of heterogeneous graphs. Thus, Equation (1) can be more precisely expressed as:

$$e_{ij}^m = \sigma(a_m^T [h_i || h_j]), \tag{2}$$

where  $h_i$  denotes the features vector of node  $v_i$ ;  $\sigma$  denotes the activation function (LeakyReLU was selected in this case);  $a_m$  denotes the node-level attention vector based on the motif subgraph,  $\mathcal{G}_m$ ; and  $||$  denotes the concatenate vector operation. After obtaining the initial attention scores of all the first-order neighbors of node  $v_i$ , normalization is undertaken using

a SoftMax function, and this gives the attention weights,  $\alpha_{ij}$ , for the degree of importance,  $e_{ij}$ , from node  $v_j$  to node  $v_i$ :

$$\alpha_{ij}^m = \frac{\exp(\sigma(a_m^T[h_i||h_j]))}{\sum_{n \in \mathcal{N}_i^m} \exp(\sigma(a_m^T[h_i||h_n]))}, \quad (3)$$

where  $\mathcal{N}_i^m$  denotes the neighboring nodes of node  $v_i$  in  $\mathcal{G}_m$ . It should be noted that  $\alpha_{ij}$  is not symmetric, so nodes  $v_i$  and  $v_j$  do not contribute to each other equally. This is not only because of the order of the vector concatenation on the numerator in Equation (3), but also because they have different neighbors. The embedding vector of node  $v_i$  in the motif subgraph  $\mathcal{G}_m$  is now an attention-based weighted aggregation of its neighboring nodes' features:

$$z_i^m = \sigma \left( \sum_{j \in \mathcal{N}_i^m} \alpha_{ij}^m h_j \right), \quad (4)$$

As the attention weight,  $\alpha_{ij}^m$ , is based on the specific motif subgraph,  $\mathcal{G}_m$ , it reflects only the side profile of the node representations in a specific motif pattern. This is also because the non-Euclidean spatial properties of graph data, especially heterogeneous graph data, exhibit high levels of variance [19]. To enrich the ability of the model and stabilize the training process, MBHAN extends the node-level attention to multi-head attention so as to avoid overfitting. To do this, MBHAN iterates the node-level attention training process  $K$  times and concatenates the learned representations to give the specific embedding:

$$z_i^m = \big\|_{k=1}^K \sigma \left( \sum_{j \in \mathcal{N}_i^m} \alpha_{ij}^m \cdot h_j \right), \quad (5)$$

Given a heterogeneous graph,  $\mathcal{G}$ , the set of its motif subgraphs ( $\mathcal{G}_{m_0}, \mathcal{G}_{m_1}, \dots, \mathcal{G}_{m_k}$ ) can be extracted easily after determining its network connectivity pattern. By inputting the features of the nodes, MBHAN can obtain the set of node representation vectors corresponding to all the motif subgraphs in the node-level attention neural network, ( $\mathcal{Z}_{m_0}, \mathcal{Z}_{m_1}, \dots, \mathcal{Z}_{m_k}$ ).

#### 4.2. Motif Subgraph-Level Attention Mechanism

In heterogeneous graphs, each node consists of multifaceted semantic information, but a particular motif subgraph only reflects a side view of that node given its current motif connectivity pattern. To more thoroughly learn the embedding of the nodes, it is necessary to fuse the representations of the corresponding nodes in all the motif subgraphs. To achieve this, MBHAN employs a motif subgraph-level semantic attention mechanism for each possible triangular motif subgraph in the heterogeneous graph. This enables it to learn the importance of different motif subgraphs for the final representation of the node and fuse them within an end-to-end learning process. Using the node embeddings learned in the node-level attention mechanism as input, the learning process for each motif subgraph can be represented as follows:

$$(\beta_{m_0}, \beta_{m_1}, \dots, \beta_{m_k}) = att_{motif-level}(\mathcal{Z}_{m_0}, \mathcal{Z}_{m_1}, \dots, \mathcal{Z}_{m_k}), \quad (6)$$

where  $att_{motif-level}$  refers to the execution of a motif subgraph-level deep learning attention process. This can capture the importance of all the triangular motif subgraphs in the heterogeneous graph for the final embeddings of the nodes. To capture the impact of each motif subgraph on the final node representation, MBHAN first uses a one-layer multilayer perceptron (MLP) to nonlinearly transform the embeddings of the nodes in the corresponding motif subgraphs. It then measures the motif importance of the impact on

the final node embedding by using the motif semantic-level attention vector,  $q$ , where  $W$  is the weight matrix and  $b$  is the bias vector:

$$w_{m_i} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} q^T \cdot \tanh(W \cdot z_i^{m_i} + b), \tag{7}$$

Note that MBHAN attempts to learn the representation of all the nodes in the heterogeneous graphs, so a simple weighted global average for the different types of nodes is clearly inappropriate. As the motif subgraph division of MBHAN is predicated on the heterogeneity of the nodes, different motif subgraphs will have semantic nuances due to the node types, and their influence on the node-level representation learning process will certainly vary. So, to isolate the influence of node heterogeneity, different types of nodes should be treated differently in the attention mechanism’s execution. Equation (7) can therefore be reformulated as:

$$w_{m_i^t} = \frac{1}{|\mathcal{V}^t|} \sum_{i \in \mathcal{V}^t} q^T \cdot \tanh(W \cdot z_i^{m_i} + b), t \in \mathcal{T}, \tag{8}$$

where  $w_{m_i^t}$  is the importance coefficient for the  $t$ -type nodes in the motif subgraph,  $\mathcal{G}_{m_i}$ , and  $\mathcal{V}^t$  is the set of  $t$ -type nodes. To ensure the stability of the training process, different types of nodes all share the one-layer MLP parameters described above when calculating each separate attention coefficient. Having calculated the importance coefficients for different types of nodes in different motif subgraphs, they are normalized to obtain the motif subgraph-level attention weights for each node type,  $t$ , in  $\mathcal{G}_{m_i}$  by using a SoftMax function:

$$\beta_{m_i^t} = \frac{\exp(w_{m_i^t})}{\sum_{t=1}^{\mathcal{T}} \exp(w_{m_i^t})}, \tag{9}$$

$\beta_{m_i^t}$  can be interpreted as the contribution of each of the  $t$ -type nodes in  $\mathcal{G}_{m_i}$  to a specific task. The higher the value of  $\beta_{m_i^t}$ , the more important the  $t$ -type nodes in  $\mathcal{G}_{m_i}$ . The importance varies across different tasks. By fusing the various motif-level weights as coefficients, the final  $t$ -type nodes embedding will be:

$$\mathcal{Z}^t = \sum_{i=1}^{\mathcal{M}} \beta_{m_i^t} \cdot \mathcal{Z}_{m_i}^t, \tag{10}$$

Unlike the Heterogenous Graph Attention Network (HAN) described in [19], which averages all meta-path-based node embeddings, MBHAN uses a vector concatenation strategy to accentuate the different influences of different motif subgraphs on the final node embeddings. As a result, the node representations in different motif subgraphs are reflected in the attention deep learning process at the motif subgraph-level. For node types that do not exist in the motif subgraphs (e.g., nodes of type  $D$  in the “ $A - C - B$ ” motif subgraph in Figure 2), the final global node representation,  $\mathcal{Z}$ , of the heterogeneous graph employs zero-complement and feature alignment operations.

As shown in Figure 3, the final node representations consist of an aggregation of all the motif-specific subgraph semantics, for which different loss functions can be designed to apply the representations to different downstream tasks. For node classification tasks, MBHAN uses a cross-entropy loss function:

$$Loss = - \sum_{l \in \mathcal{Y}_L} Y^l \ln(C \sqcup \mathcal{Z}^l), \tag{11}$$

where  $C$  is the classifier parameters;  $\mathcal{Y}_L$  is the set of nodes with labels; and  $Y^l$  and  $\mathcal{Z}^l$  are the labels and embeddings of the labeled nodes, respectively. Guided by the labeled node data, MBHAN can use back propagation to optimize the model and learn the embeddings of the nodes.

### 4.3. Node Feature Mapping Mechanism

Having presented the details of the model for deep learning of the node-level and motif subgraph-level attention mechanisms, we now need to consider the incompatibility of feature spaces due to the heterogeneity of the nodes. By seeking to preserve the rich semantics contained in different types of nodes and their links during the graph representation learning process, we are obliged to deal with there being different feature spaces for different types of nodes. Many meta-path-based heterogeneous graph representation learning methods transform heterogeneous graphs into homogeneous graphs for analysis [19,20]. This results in a loss of semantic information for the non-end nodes in the meta-paths. MBHAN aims to learn the embedding of global nodes. In the motif subgraphs subdivided by node types, the node-level attention learning process usually aggregates neighborhood information that is different from the current type of node. In real-world graph datasets, it is also quite possible that only features of specific types of nodes can be accurately extracted. In the DBLP heterogeneous graph dataset shown in Figure 1a, for instance, only features of Paper-type nodes (bag-of-words vectors) can be observed, while features relating to the Author, Term, and Conference-type nodes cannot be easily obtained.

To address the above problem, one can create a matrix,  $M_{\perp_i}$ , that is specific to the type transformation to project the features of different types of nodes into the same feature space. The feature conversion process for node  $v_i$  can be formulated as follows:

$$h'_i = M_{\perp_i} \cdot h_i, \quad (12)$$

where  $h_i$  and  $h'_i$  are the original and transformed features of node  $v_i$ . In MBHAN, this type of transformation is implemented by employing a shallow perceptron whose output dimensionality is consistent with that of the node features that already exist. The specific feature transformation matrix parameters of the model can be guided and learned according to particular downstream tasks throughout the training process. Note that, when undertaking a feature transformation process for multiple types of nodes, the perceptron models should be segregated from each other, i.e.,  $M_t \neq M_{t'}$ . We expect this type of transformation process to only serve for specific node types, where it is important to avoid any possible interference between different types of nodes.

Although Equation (12) provides the process for feature transformation, it does not address the problem of missing node features in heterogeneous graphs. Looking at graph representation learning methods based on a random walk strategy [24,43,44], there is an essential assumption being made: that the rich semantic interactions between node pairs will be reflected in the structure of the heterogeneous graphs. So, if an author has numerous important paper nodes connected to him (her) in the field of data mining, it means that the author has contributed significantly to the field of data mining, and vice versa. If this assumption is correct, the node features obtained by graph analysis from the node connectivity patterns can describe the knowledge contained in the nodes from another perspective (i.e., a graph structure perspective). A classic “random walk + Skip-Gram” strategy is therefore typically included in the heterogeneous graph feature preprocessing process. In this paper, MBHAN adopts the approach of Grover et al. [45], where parametrically controlled node sequence selection is applied during the random walk process:

$$\alpha_{pq}(s, c) = \begin{cases} \frac{1}{p}, & \text{if } d_{sx} = 0 \\ 1, & \text{if } d_{sx} = 1 \\ \frac{1}{q}, & \text{if } d_{sx} = 2 \end{cases} \quad (13)$$

where  $s$  is the previous hop node;  $c$  is the current node;  $\alpha_{pq}$  is the transfer probability;  $d_{sx}$  is the shortest path distance between node  $s$  and node  $c$ 's neighboring node,  $x$ ;  $p$  is the return probability parameter, which corresponds to the Breadth First Search (BFS) of the walk process; and  $q$  is the departure probability parameter, which corresponds to the Depth First Search (DFS). This approach can adapt the random walk strategy for a specific downstream task or a specific graph structure. In the experiments presented in Section 5, we

make  $p = q = 1$ . This enables a fair comparison between MBHAN and other benchmark algorithms and avoids the need to set specific parameters for the datasets. Doing this collapses the node sequence generation process into a classic random walk process [43]. Finally, the sequence of nodes is fed into a Skip-Gram model to obtain the embedding of the nodes. This is treated as a “structural” feature,  $h^{str}$ , and the corresponding inherent features of the nodes themselves are treated as “semantic” features labeled  $h_i^{sem}$ . Equation (12) can then be reformulated as follows:

$$h_i^{sem} = M_{t_i} \cdot h^{str}, \tag{14}$$

For any given heterogeneous graph, the input layer of the hierarchical attention model accepts nodes with “semantic” features directly, while nodes without intrinsic features have to pass through a random walk model and undergo the “structural” to “semantic” feature transformation described by Equation (14).

The overall MBHAN process is shown in Algorithm 1. In summary, MBHAN provides an end-to-end semi-supervised node representation learning approach that includes a feature space mapping process, a node-level attention mechanism, and a motif subgraph-level attention mechanism. For specific downstream tasks, MBHAN can automatically learn the relevant feature transformation parameters and node-level and motif subgraph-level attention weight parameters. In the next section, we report on an evaluation of the full-scale performance of MBHAN when handling a node classification task and a clustering task. We then analyze the impact of various hyperparameters on its performance.

---

**Algorithm 1** The Overall MBHAN Process

---

- 1: **Input:** A heterogeneous graph:  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ;
  - 2: Node types:  $\mathcal{T}$ ;
  - 3: A random walk + Skip-Gram Parameters set:  $c$ ;
  - 4: The inherent semantic features contained in some types of nodes:  $h^{sem}$ ;
  - 5: The collection of motif subgraphs with node-type connectivity patterns based on the heterogeneous graph,  $\mathcal{G}$ :  $(\mathcal{G}_{m_0}, \mathcal{G}_{m_1}, \dots, \mathcal{G}_{m_k})$ ;
  - 6: The number of attention heads:  $K$ ;
  - 7: **Output:** Node representation learning vectors:  $\mathcal{Z}$ ;
  - 8: Generate node “structure” features using the random walk strategy;
  - 9:  $h^{str} \leftarrow$  Random Walk, ( $c$ );
  - 10: **for** every node type,  $t$ , in  $\mathcal{T}$  **do**:
  - 11:     **if**  $v_i^t$  does not have semantic features  $h^{sem}$  **do**;
  - 12:          $h_i^{sem} = M_t \cdot h^{str}$ ;
  - 13: Integration of all node features  $\rightarrow h$ ;
  - 14: **for** every motif sub-graph,  $\mathcal{G}_m$ , in  $(\mathcal{G}_{m_0}, \mathcal{G}_{m_1}, \dots, \mathcal{G}_{m_k})$ :
  - 15:     **for**  $k = 1 \dots k$  **do**;
  - 16:         **for**  $v_i$  in  $\mathcal{G}_m$  **do**;
  - 17:             **find, for**  $v_i$ , the neighboring nodes set,  $\mathcal{N}_i^m$ ;
  - 18:             **for**  $j \in \mathcal{N}_i^m$  **do**;
  - 19:                 Calculate the weight coefficient  $\alpha_{ij}^m$ ;
  - 20:                 Calculate the motif subgraph-specific node embedding;
  - 21:                  $z_i^m \leftarrow \sigma(\sum_{j \in \mathcal{N}_i^m} \alpha_{ij}^m \cdot h_j)$ ;
  - 22:             Calculating embeddings learned from all attention mechanisms;
  - 23:              $z_i^m \leftarrow \frac{K}{K=1} \parallel \sigma(\sum_{j \in \mathcal{N}_i^m} \alpha_{ij}^m \cdot h_j)$ ;
  - 24: Calculate the weight of the motif subgraphs,  $\beta_{m_i}^t$ ;
  - 25: Calculate the  $t$ -type node embedding  $\mathcal{Z}^t \leftarrow \frac{M}{M=1} \parallel \beta_{m_i}^t \cdot \mathcal{Z}_{m_i}^t$ ;
  - 26: Fuse all types of node embedding,  $\mathcal{Z}$ ;
  - 27: Calculate the downstream task-specific loss functions;
  - 28: Back-propagate and update the parameters in MBHAN;
  - 29: **return**  $\mathcal{Z}$ .
-

### 5. Experiments

Before reporting on our evaluation of MBHAN’s performance, we will outline the two heterogeneous graph datasets and the state-of-the-art benchmark heterogeneous graph representation learning methods that were used in our experiments.

#### 5.1. Datasets

The experiments were performed using two real-world heterogeneous graph datasets, which were often used as the benchmark datasets to evaluate the performance of the proposed methods [19,36,46]. Table 2 shows their key statistics, including all the possible triangular motif patterns and details of the nodes and links for the corresponding subgraphs.

**Table 2.** Details of the datasets used in the MBHAN evaluation experiments.

	Nodes Type	Nodes Number	Is Labeled	Is Featured	Motif Pattern				
DBLP_ four_area	Author (A)	14475	•	○					
	Paper (P)	14376	•	•					
	Term (T)	8920	○	○					
	Conference (C)	20	○	○					
	Nodes: 26478	14396	0	28871	20175	23316	37771	18852	23282
	Links: 39860	14376	0	56170	33720	129000	156418	110187	114615
ACM	Author (A)	7167	○	○					
	Paper (P)	4025	•	•					
	Subject (S)	60	○	○					
	Venue (V)	73	○	○					
	Nodes: 10871	4097	0	11259	6089	4158	11246	4072	0
	Links: 13092	4024	0	17426	8304	8050	17426	4012	0

DBLP\_four\_area [47]: is a subset of the academic citation heterogeneous network DBLP (<https://dblp.uni-trier.de>). The database covers four domains: databases; data mining; information retrieval; and artificial intelligence. It contains four different types of nodes (Author, Paper, Term, and Conference) and three different types of links (Author↔Paper, Paper↔Term, and Paper↔Conference). The features of the Paper type nodes consist of their keyword bag-of-words vectors. The features of the Author type nodes are a composite of the bag-of-words vectors for all the Paper type nodes connected to this type of node. The Term type nodes and Conference type nodes do not have any features. The Paper type nodes in the dataset are labeled according to the research fields that correspond to their publication sites. The Author type nodes are labeled according to the research fields associated with their published papers. A graph schema of the dataset is shown in Figure 1a.

ACM (<http://dl.acm.org/>): Papers published in KDD, SIGMOD, SIGCOMM, MobiCOMM, and VLDB were extracted and grouped into three research areas: databases; wireless communication; and data mining. This dataset contains four different types of nodes (Author, Subject, Paper, and Venue) and three different types of links (Author↔Paper, Paper↔Subject, and Paper↔Venue). We used a dataset preprocessing approach similar to

that used in [48], where the features of the Paper type nodes were taken to be the vectors of the bag-of-words elements given by their keywords. They were labeled according to the conference in which the papers were published. In this dataset, the Subject type nodes and Venue type nodes do not have any node features. A graph schema for this dataset is shown in Figure 1b.

### 5.2. Baseline Algorithms

MBHAN was compared with the following baseline algorithms, which together cover homogeneous graph representation learning methods, heterogeneous graph representation learning methods, and graph neural networks.

DeepWalk [43]: is a graph representation method based on a random walk strategy that is usually applied to homogeneous graphs. In the experiments, it treated the datasets as homogeneous graphs because it could not handle the heterogeneity of the nodes.

metapath2vec [44]: is a heterogeneous graph representation learning method based on a meta-path random walk that leverages Skip-Gram to learn the embedding of the nodes. In the experiments, all possible meta-paths consisting of three node types were processed. The reported results give the average performance.

GCN [9]: is a semi-supervised graph convolutional neural network designed for homogeneous graphs. As not all nodes in the datasets have features, the experiments used a meta-path based heterogeneous graph transformation operation. For the Author node classification and clustering task in the DBLP\_four\_area dataset, the meta-paths  $A - P - A$ ,  $A - P - C - P - C - A$ , and  $A - P - T - P - A$  were employed for the transformation. For the Paper node classification and clustering task, the meta-paths  $P - A - P$ ,  $P - C - P$ , and  $P - T - P$  were employed, with the given results being the average performance. For the ACM dataset, the meta-paths  $P - A - P$  and  $P - S - P$  were employed, with the average performance again being reported.

HAN [19]: is a hierarchical graph attention neural network that can be used for heterogeneous graphs. It takes into account both node-level attention and meta-path-level attention. We adopted the meta-path selection scheme given in the literature. The meta-paths  $A - P - A$ ,  $A - P - C - P - C - A$ , and  $A - P - T - P - A$  were used for the Author node classification and clustering task in the DBLP\_four\_area dataset, while the meta-paths  $P - A - P$ ,  $P - C - P$ , and  $P - T - P$  were used for the Paper node classification and clustering task. For the ACM dataset, the meta-paths  $P - A - P$  and  $P - S - P$  were used.

GTN [21]: is a heterogeneous graph neural network representation learning method that requires no prior knowledge. It generates new graph structures from multiple candidate adjacency matrices from the original graph to achieve more efficient graph convolution operations.

MBHAN<sub>non\_type</sub>: is a simplified version of MBHAN that does not distinguish between the node types when calculating the motif subgraph attention. To be precise, after calculating the importance coefficients for the different motif subgraphs in Equation (7), a SoftMax function is applied to normalize all the node types.

### 5.3. Implementation Details

For the full version of MBHAN, we first randomly initialized the parameters and optimized the model using Adam [49]. The learning rate was set at 0.005, the weight decay parameter was 0.001, the dimensionality of the motif subgraph-level attention vector,  $q$ , was 128, the number of multi-headed attention mechanisms,  $K$ , was set at 8, and the dropout ratio [50] was set at 0.6. To ensure that all of the experiments were fair, the semi-supervised graph neural network models, i.e., GCN, GAT, HAN, and GTN used exactly the same split for their training, validation, and test sets (80%: 10%: 10%). For the random walk-based graph representation learning methods, i.e., DeepWalk and metapath2vec, the Skip-Gram model context window size was set at 5, the walk sequence length was set at 100, the number of walks per node was set at 5, the negative sampling dimension was set at 5, and the embedding vector dimension was set at 128. All of the experiments were executed

on a Lenovo R9000P 2021 with an AMD 3.2GHz processor, 64Gb RAM, and an NVIDIA RTX3060 laptop graphics card with a video memory capacity of 6Gb.

#### 5.4. Multi-Class Classification

Multi-class nodes are nodes' labels with more than two classes, but where each node is assigned to only one label in the graph. The multi-class classification prediction assigns one and only one label to each node. MBHAN uses a fully connected linear layer to perform the multi-class node classification task. We employed a 10-fold cross-validation experiment and report the average Macro-F1 and Micro-F1 performance for MBHAN and all the other baseline methods in Table 3.

**Table 3.** Node classification performance results (%).

Datasets	DBLP_Four_Area				ACM	
Target Nodes	Author		Paper		Paper	
Metrics	Mac-F1	Mic-F1	Mac-F1	Mic-F1	Mac-F1	Mic-F1
DeepWalk	85.01	85.71	84.35	84.17	84.71	85.07
metapath2vec	87.74	87.58	84.61	84.89	84.81	84.57
GCN	86.47	87.09	85.84	86.58	85.77	85.29
GAT	88.57	89.11	87.85	89.64	87.44	87.80
HAN	93.08	93.99	89.48	92.40	90.40	90.72
GTN	94.14	94.17	88.80	91.64	90.77	90.68
MBHAN <sub>non_type</sub>	94.47	94.14	<b>90.00</b>	93.43	90.71	90.32
MBHAN	<b>95.45</b>	<b>95.57</b>	<b>90.00</b>	<b>93.65</b>	<b>91.17</b>	<b>90.82</b>

It can be seen that MBHAN outperformed the various benchmark methods. Out of the traditional graph representation learning methods, metapath2vec with guided meta-paths guided performed better than DeepWalk. This again confirms that considering the heterogeneity of the nodes preserves more knowledge in the graph. The graph neural network-based methods, e.g., GAT and GCN, not only retained the structural information for the graph, but also attempted to fuse the node features. These methods performed better than the traditional graph representation learning methods. Looking more closely at the results, because of its differential treatment of the nodes' neighboring objects, GAT was better able to capture the degree of importance of the node neighbors in the graph than the simple GCN approach, where the neighboring nodes are merely averaged. When compared with GAT, HAN not only focused on relevant knowledge of the nodes and their neighbors, but also differentiated the influence of substructures (subgraphs that satisfied the meta-paths) on the final embedding of the nodes from a high-dimensional (meta-path) perspective. Unfortunately, it remains the case that HAN transforms heterogeneous graphs into homogeneous ones, so there was inevitably some loss of information and fluctuation in the effectiveness of the artificially selected meta-paths for the downstream tasks. Thus, GTN's convolution of multiple meta-path graphs enabled it to learn the importance of the length of the meta-paths more adaptively and gain a better performance than HAN.

Unlike GTN, MBHAN not only has the advantages of the HAN hierarchical attention mechanism, but also avoids the need for homogeneous transformation of heterogeneous graphs by using motif subgraphs, so it does not require a priori meta-path knowledge. MBHAN can also classify different types of nodes, such as the Author node and Paper node in the DBLP\_four\_area dataset, simultaneously without needing to change the graph neural network model. So, to sum up, MBHAN performed better than GTN because it can handle non-Euclidean spatially heterogeneous graph data with high degrees of variance and because using a hierarchical attention mechanism enables it to better capture the differentiation between nodes than other graph neural network methods. As a final observation, note that the performance of MBHAN was also better than the simplified MBHAN<sub>non\_type</sub>. This indicates that having different types of nodes in downstream tasks can mean that

different motif subgraphs are of different importance, so being able to distinguish different node types makes it possible to capture this subtle differential knowledge more precisely.

### 5.5. Clustering

For the clustering performance evaluation, MBHAN used normalized mutual information values (NMI). To this end, a  $k$ -means algorithm was employed in the experiments to cluster the final embedding of the nodes, and the number of clusters,  $K$ , was set to the number of classes. As the performance of a  $k$ -means algorithm is influenced by the initial cluster centers, we repeated the experiment 10 times and report here the average performance.

It can be seen from Table 4 that MBHAN performed better than all the baseline methods. By fusing the inherent features of the nodes in the graph, the graph neural network-based methods generally performed better than the traditional random walk-based graph representation learning methods. As GCN did not distinguish the importance of neighboring nodes, in most cases its performance was inferior to GAT. This underscores the fact that attention mechanisms capture more meaningful node embeddings in the graph neural network representation learning process.

**Table 4.** Node clustering performance results (%).

Datasets	DBLP_Four_Area		ACM
Metrics	NMI		NMI
Target Nodes	Author	Paper	Paper
DeepWalk	73.49	57.91	43.89
metapath2vec	66.17	53.48	22.47
GCN	76.31	60.47	54.26
GAT	76.80	60.70	58.47
HAN	79.87	62.79	61.56
GTN	80.05	63.09	61.77
MBHANnon_type	81.17	63.00	60.74
MBHAN	<b>81.75</b>	<b>63.14</b>	<b>61.79</b>

Note, however, that HAN, which does employ a hierarchical attention mechanism, performed less well than GTN, where there was no use of prior knowledge, and MBHAN. This is because it focused only on the importance of certain meta-paths and inevitably missed information about the influence of other high-dimensional node sets on the final embedding.

### 5.6. Comparative Statistical Tests for the Different Algorithms

The previous two subsections involved the comparison of multiple algorithms on multiple datasets. However, the same algorithm may not have the same ranking on different datasets. We therefore undertook some statistical tests to assess the overall performance of each particular algorithm across the multiple datasets.

A Friedman test [51] can be employed to determine whether the performance of a particular algorithm (MBHAN, in this case) is significantly different from other algorithms. It is a classic non-parametric statistical test that is based on the hypothesis that there is no significant difference in the overall distribution of multiple pairs of algorithms' mean ranks. If we let  $Rank_j^i$  be the rank of the  $j$ -th method on the  $i$ -th data set, its mean performance ranking on a dataset can be calculated using Equation (15), where  $N$  is the number of methods to be compared and  $S$  is the number of datasets. The Friedman statistic can then be computed using Equation (16). As  $\tau_{\chi^2}$  was found to be too conservative, an improved version is given by Equation (17).

$$Rank_j = \frac{1}{S} \sum_{i=1}^S Rank_j^i, j \in [1, N], \quad (15)$$

$$\tau_{\chi^2} = \frac{12S}{N(N+1)} \left( \sum_{i=1}^N Rank_i^2 - \frac{N(N+1)^2}{4} \right) \tag{16}$$

$$F = \frac{(S-1)\tau_{\chi^2}}{S(N-1) - \tau_{\chi^2}}, \tag{17}$$

To perform an overall comparison, the experimental results for the different datasets need to be combined.  $S$  was set at 3 and  $N$  was set at 8. The Mac-F1, Mic-F1, and NMI Friedman test values were 41.200, 24.999, and 73.600, respectively. These are all much higher than the critical value (2.76) for a significance level of  $\alpha = 0.05$ . Therefore, the original hypothesis can be safely rejected. In other words, there are significant performance differences between the different methods.

Because the null-hypothesis is rejected, a post-hoc test can be proceeded. The Nemenyi test [51], which is calculated using Equation (18), defines the critical difference ( $CD$ ) value for a case where two methods are significantly different with a certain confidence  $(1 - \gamma)$ . Here,  $q_{\gamma}$  is the critical value based on the Studentized range statistic divided by  $\sqrt{2}$ .

$$CD = q_{\gamma} \sqrt{\frac{N(N+1)}{6S}}, \tag{18}$$

The critical difference for the 95% confidence interval was calculated ( $CD = 6.06$ ). The results of the test are shown in Figure 4, where the mean rank of each method is marked by a dot, and the length of the horizontal bar across each dot shows the critical difference. For any two methods, a method outperforms another one if its mean rank is smaller. More strictly, if the mean ranks of any two methods differ by more than  $CD$ , the superiority is significant. According to the mean ranks and the results of the Nemenyi test, we can conclude that our MBHAN method achieves the best overall performance, and significantly outperforms some of these methods.

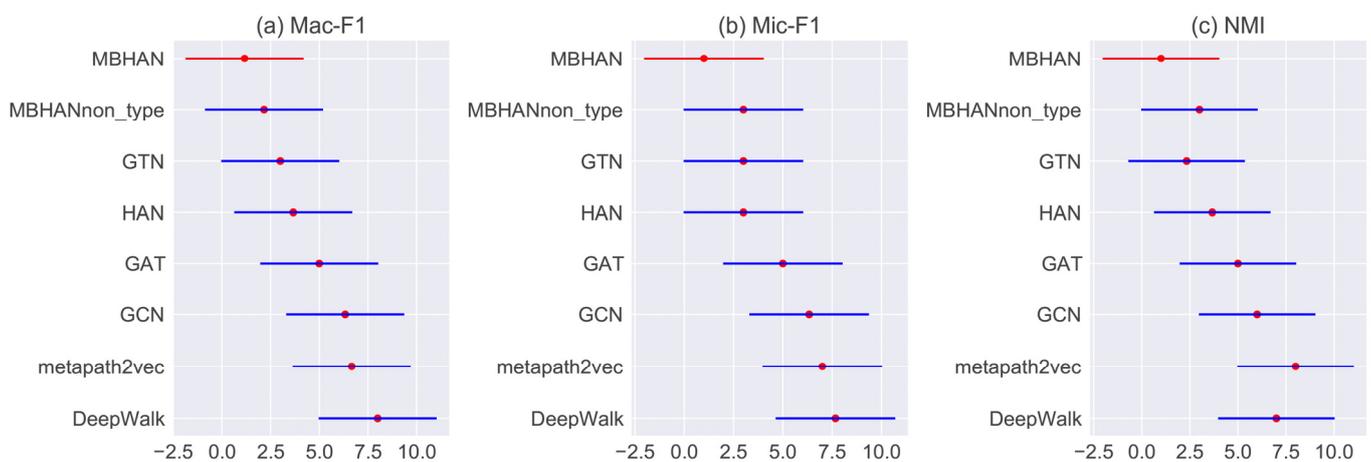


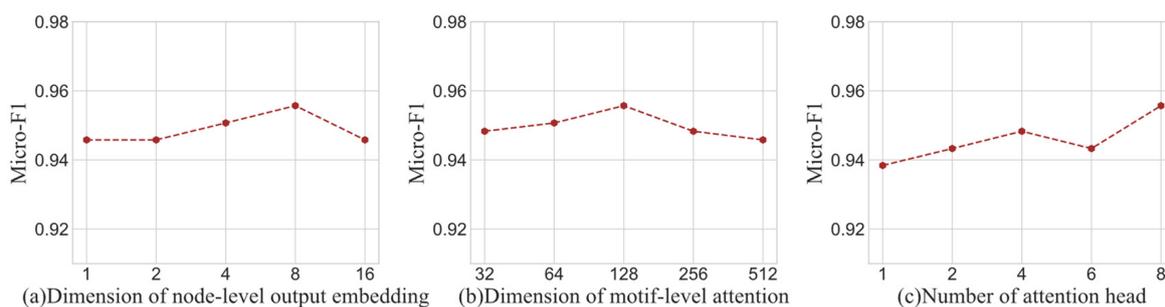
Figure 4. Average rank results of Nemenyi test.

### 5.7. Analysis of the Hyperparameters

The parameter sensitivity was analyzed by using Micro-F1 metrics for MBHAN’s classification performance in relation to the Author type nodes in the DBLP\_four\_area dataset. The analyzed parameters included the embedding dimensions of the node-level attention mechanism output, the motif subgraph-level attention vector dimensions,  $q$ , and the multi-head attention parameter,  $K$ .

**Node-level attention mechanism output dimensions:** As the final node embedding of MBHAN was a concatenation of the outputs from the motif subgraphs’ attention process, its size was impacted by the number of motif subgraphs that could be subdivided by the specific heterogeneous graphs. The results are shown in Figure 5a, where it can be seen

that its performance initially improved as the number of embedding dimensions increased, but then began to slowly decrease. This is due to the fact that the final representation of the nodes requires a suitable dimension for encoding. Furthermore, when the output dimensions of the node-level attention process became too large and multiple motif subgraphs with various kinds of representations had to be stacked, this introduced redundancy into the node embedding. Note, however, that the dimensions of the node-level attention output did not cause significant performance fluctuations in the downstream tasks because MBHAN not only learned the attention weights at the node level and motif subgraph level, but also the differentiation between different types of heterogeneous graph node. Thus, the concatenation strategy for the final embedding of the nodes could assign useful knowledge to the representation feature space.



**Figure 5.** MBHAN parameter sensitivity analysis.

**The motif subgraph-level attention vector,  $q$ :** The results for how the motif subgraph-level attention mechanism was affected by the dimensionality of vector  $q$  are shown in Figure 5b. Here, it can be seen that MBHAN's performance increased in line with the dimensions of  $q$  and reached its best performance at 128. After that, its performance started to decline because of overfitting of the training process caused by the large number of dimensions.

**The multi-head attention parameter,  $K$ :** The influence of the multi-head attention parameter,  $K$ , on MBHAN's performance is shown in Figure 5c. When  $K = 1$ , MBHAN did not adopt the multi-head attention strategy. Its performance improved slightly as the number of heads increased. This is because the multi-head attention strategy essentially involves integrating several independent attention coefficient computations. This strategy not only characterizes the nodes from multiple perspectives, but also prevents overfitting and makes the training process more stable.

## 6. Conclusions

In this paper, we have proposed a method for motif-based hierarchical attentional graph neural network representation learning, called MBHAN. It consists of a node-level attention mechanism and a motif semantic attention mechanism. MBHAN does not require any prior knowledge, but instead seeks to reflect the node features from different perspectives. MBHAN treats the relative importance of various node types differently for the node embedding during the learning process. This enables it to capture subtle nuances caused by having different node types in the motif subgraphs. A full-scale evaluation of the performance of MBHAN was undertaken on two heterogeneous graph datasets, which involved node classification and clustering tasks, and the impact of various hyperparameters on its performance were also evaluated. The F1 and NMI metrics of the results and a statistical analysis show that the proposed method can outperform other state-of-the-art methods.

It should be noted that the feature space mapping proposed by MBHAN is still not able to completely solve the problem of there being incompatible feature spaces for different node types in heterogeneous graphs. This will therefore be the focus of our future research.

**Author Contributions:** Conceptualization, W.L. and M.T.; methodology, Q.H., W.L., and M.T.; software, Q.H. and J.J.; data curation, Q.H. and J.J.; writing—original draft, Q.H.; writing—review and editing, W.L. and M.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Guizhou Normal University 2021 Provincial Teaching Content and Curriculum Reform Foundation (grant number 2021047).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zhou, T.; Qi, S.; Wang, W.; Shen, J.; Zhu, S.C. Cascaded Parsing of Human-Object Interaction Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 2827–2840. [[CrossRef](#)] [[PubMed](#)]
2. Zhou, T.; Li, L.; Li, X.; Feng, C.M.; Li, J.; Shao, L. Group-Wise Learning for Weakly Supervised Semantic Segmentation. *IEEE Trans. Image Process.* **2022**, *31*, 799–811. [[CrossRef](#)] [[PubMed](#)]
3. Li, Y.; Zemel, R.; Brockschmidt, M.; Tarlow, D. Gated Graph Sequence Neural Networks. In Proceedings of the ICLR'16, San Juan, Puerto Rico, 2–4 May 2016.
4. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The Graph Neural Network Model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80. [[CrossRef](#)] [[PubMed](#)]
5. 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.
6. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; van den, B.R.; Titov, I.; Welling, M. *Modeling Relational Data with Graph Convolutional Networks*; Springer: Cham, Switzerland, 2018.
7. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Curran Associates Inc.: Barcelona, Spain, 2016; pp. 3844–3852.
8. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Long Beach, CA, USA, 2017; pp. 1025–1035.
9. Kipf, T.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907.
10. Ashish, V.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Long Beach, CA, USA, 2017; pp. 6000–6010.
11. Dzemir, B.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.
12. Kelvin, X.; Ba, J.L.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R.S.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; JMLR.org: Lille, France, 2015; Volume 37, pp. 2048–2057.
13. Zhixing, T.; Wang, M.; Xie, J.; Chen, Y.; Shi, X. Deep Semantic Role Labeling with Self-Attention. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
14. Patrick, V.; Strubell, E.; McCallum, A. Simultaneously Self-Attending to All Mentions for Full-Abstract Biological Relation Extraction. *arXiv* **2018**, arXiv:1802.10569.
15. Zichao, Y.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E.H. Hierarchical Attention Networks for Document Classification. In Proceedings of the NAACL, San Diego, CA, USA, 12–17 June 2016.
16. Petar, V.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2018**, arXiv:1710.10903.
17. Zhouhan, L.; Feng, M.; dos Santos, C.N.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A Structured Self-Attentive Sentence Embedding. *arXiv* **2017**, arXiv:1703.03130.
18. Yizhou, Z.; Xiong, Y.; Kong, X.; Li, S.; Zhu, Y. Deep Collective Classification in Heterogeneous Information Networks. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018.
19. Xiao, W.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; Yu, P.S. Heterogeneous Graph Attention Network. In *The World Wide Web Conference*; Association for Computing Machinery: San Francisco, CA, USA, 2019; pp. 2022–2032.
20. Shaohua, F.; Zhu, J.; Han, X.; Shi, C.; Hu, L.; Ma, B.; Li, Y. Metapath-Guided Heterogeneous Graph Neural Network for Intent Recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 4–8 August 2019; Association for Computing Machinery: Anchorage, AK, USA, 2019; pp. 2478–2486.
21. Seongjun, Y.; Jeong, M.; Kim, R.; Kang, J.; Kim, H.J. Graph Transformer Networks. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.

22. Carranza, A.G.; Rossi, R.A.; Rao, A.; Koh, E. Higher-Order Clustering in Complex Heterogeneous Networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 6–10 July 2020; Association for Computing Machinery: San Francisco, CA, USA, 2020; pp. 25–35.
23. Yang, C.; Liu, M.; Zheng, V.W.; Han, J. Node, Motif and Subgraph: Leveraging Network Functional Blocks through Structural Convolution. In Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, Spain, 28–31 August 2018.
24. Qian, H.; Lin, F.; Wang, B.; Li, C. Mbrep: Motif-Based Representation Learning in Heterogeneous Networks. *Expert Syst. Appl.* **2021**, *190*, 116031.
25. Daredddy, 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.
26. Li, Y.; Lou, Z.; Shi, Y.; Han, J. Temporal Motifs in Heterogeneous Information Networks. In Proceedings of the 14th International Workshop on Mining and Learning with Graphs (MLG), London, UK, 20 August 2018.
27. 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](#)]
28. Lei, X.U.; Huang, L.; Wang, C. Motif-Preserving Network Representation Learning. *J. Front. Comput. Sci. Technol.* **2019**. [[CrossRef](#)]
29. Yu, Y.; Lu, Z.; Liu, J.; Zhao, G.; Wen, J. Rum: Network Representation Learning Using Motifs. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macau, China, 8–11 April 2019.
30. Boaz, L.J.; 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; Association for Computing Machinery: Beijing, China, 2019; pp. 499–508.
31. Hao, P.; Li, J.; Gong, Q.; Wang, S.; Ning, Y.; Yu, P.S. Graph Convolutional Neural Networks Via Motif-Based Attention. *arXiv* **2019**, arXiv:1811.08270.
32. Cai, H.; Zheng, V.W.; Chang, K.C. A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1616–1637. [[CrossRef](#)]
33. Mahmudul, H.A.S.M.; Sohel, F.; Diepeveen, D.; Laga, H.; Jones, M.G.K. A Survey of Deep Learning Techniques for Weed Detection from Images. *Comput. Electron. Agric.* **2021**, *184*, 106067.
34. Shervin, M.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; Gao, J. Deep Learning-Based Text Classification: A Comprehensive Review. *ACM Comput. Surv.* **2021**, *54*, 62.
35. Chuxu, Z.; Song, D.; Huang, C.; Swami, A.; Chawla, N.V. Heterogeneous Graph Neural Network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; Association for Computing Machinery: Anchorage, AK, USA, 2019; pp. 793–803.
36. Ziniu, H.; Dong, Y.; Wang, K.; Sun, Y. Heterogeneous Graph Transformer. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; Association for Computing Machinery: Taipei, Taiwan, 2020; pp. 2704–2710.
37. Austin, R.B.; Gleich, D.F.; Leskovec, J. Higher-Order Organization of Complex Networks. *Science* **2016**, *353*, 163–166.
38. Dawei, Z.; Zhang, S.; Yildirim, M.Y.; Alcorn, S.; Tong, H.; Davulcu, H.; He, J. A Local Algorithm for Structure-Preserving Graph Cut. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 13–17 August 2017.
39. Hao, Y.; Benson, A.R.; Leskovec, J.; Gleich, D.F. Local Higher-Order Graph Clustering. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; Association for Computing Machinery: Halifax, NS, Canada, 2017; pp. 555–564.
40. Le-kui, Z.; Yang, Y.; Ren, X.; Wu, F.; Zhuang, Y. Dynamic Network Embedding by Modeling Triadic Closure Process. In Proceedings of the AAAI, Hilton, New Orleans, 2–7 February 2018.
41. Tomas, M.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 5–10 December 2013; Curran Associates Inc.: Lake Tahoe, Nevada, 2013; Volume 2, pp. 3111–3119.
42. Jiawei, H. Mining Heterogeneous Information Networks by Exploring the Power of Links. In Proceedings of the 20th International Conference on Algorithmic Learning Theory, Porto, Portugal, 3–5 October 2009; Springer: Porto, Portugal, 2009.
43. Bryan, P.; 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; Association for Computing Machinery: New York, NY, USA, 2014; pp. 701–710.
44. Yuxiao, D.; Chawla, N.V.; Swami, A. Metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; Association for Computing Machinery: Halifax, NS, Canada, 2017; pp. 135–144.
45. Aditya, G.; Leskovec, J. Node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; Association for Computing Machinery: San Francisco, CA, USA, 2016; pp. 855–864.
46. Meng, C.; Ma, X.; Xu, M.; Wang, C. *Heterogeneous Information Network Embedding with Meta-Path Based Graph Attention Networks*; Springer: Cham, Switzerland, 2019.

47. Ming, J.; Sun, Y.; Danilevsky, M.; Han, J.; Gao, J. Graph Regularized Transductive Classification on Heterogeneous Information Networks. In Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part I, Barcelona, Spain, 20–24 September 2010; Springer: Barcelona, Spain, 2010; pp. 570–586.
48. Ziqi, L.; Chen, C.; Yang, X.; Zhou, J.; Li, X.; Song, L. Heterogeneous Graph Neural Networks for Malicious Account Detection. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, New York, NY, USA, 22–26 October 2018; Association for Computing Machinery: Torino, Italy, 2018; pp. 2077–2085.
49. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2015**, arXiv:1412.6980.
50. Nitish, S.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
51. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.