electronics

MDPI

Article

HGNN−BRFE: Heterogeneous Graph Neural Network Model Based on Region Feature Extraction

Yufei Zhao [1], Shixiao Xu [2] and Hua Duan [1,*]

[1] College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao 266590, China; yfzhao@sdust.edu.cn
[2] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China; xushixiao@sdust.edu.cn
* Correspondence: hduan@sdust.edu.cn

Abstract: With the strong capability of heterogeneous graphs in accurately modeling various types of nodes and their interactions, they have gradually become a research hotspot, promoting the rapid development of the field of heterogeneous graph neural networks (HGNNs). However, most existing HGNN models rely on meta−paths for feature extraction, which can only utilize part of the data from the graph for training and learning. This not only limits the data generalization ability of deep learning models but also affects the application effect of data−driven adaptive technologies. In response to this challenge, this study proposes a new model—heterogeneous graph neural network based on regional feature extraction (HGNN−BRFE). This model enhances performance through an "extraction−fusion" strategy in three key aspects: first, it efficiently extracts features of neighboring nodes of the same type according to specific regions; second, it effectively fuses information from different regions and hierarchical neighbors using attention mechanisms; third, it specially designs a process for feature extraction and fusion targeting heterogeneous type nodes, ensuring that the rich semantic and heterogeneity information within the heterogeneous graph is retained while maintaining the node's own characteristics during the node embedding process to prevent the loss of its own features and potential over−smoothing issues. Experimental results show that HGNN−BRFE achieves a performance improvement of 1–3% over existing methods on classification tasks across multiple real−world datasets.

Keywords: graph neural network; heterogeneous graph neural network; graph embedding; graph representation learning

check for updates

Citation: Zhao, Y.; Xu, S.; Duan, H. HGNN−BRFE: Heterogeneous Graph Neural Network Model Based on Region Feature Extraction. *Electronics* **2024**, *13*, 4447. https://doi.org/10.3390/electronics13224447

Academic Editor: M-Tahar Kechadi

Received: 15 October 2024
Revised: 7 November 2024
Accepted: 12 November 2024
Published: 13 November 2024

Copyright: © 2024 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

Many real−world datasets can be represented using heterogeneous graphs, also known as heterogeneous information networks (HINs). In these networks, nodes and edges represent different objects and their relationships, respectively. Examples of such datasets include social networks [1,2], physical systems [3], transportation networks [4], citation networks [5], recommendation systems [6], knowledge graphs [7], and more. Since heterogeneous graphs have the ability to model various types of nodes and different interactions between them, they have attracted the interest of many researchers in recent years, and heterogeneous graph neural networks (HGNNs) have been rapidly developed. Heterogeneous graph neural networks can effectively combine message passing mechanisms with complex heterogeneity so that they can well capture complex heterogeneous structures and rich semantic information [8].

Existing HGNNs are commonly studied based on meta−paths [9–13]. Meta−paths contain specific semantic information, enabling researchers to capture semantically similar nodes through these paths and obtain the final representation of the nodes [14]. However, aggregation methods in meta−path−based heterogeneous graph neural networks only

extract partial features from the surrounding area, as illustrated in Figure 1. This limitation arises from discarding some data in the graph, restricting the capabilities of heterogeneous graph neural networks. The discarded data, often close neighbors to the central node, also contain rich information crucial for enriching node features. Some models argue that a single meta−path cannot aggregate sufficient semantic information and propose the aggregation of information from multiple meta−paths to ensure nodes obtain more diverse semantics [9]. Multiple meta−paths can extract richer feature information than a single meta−path. However, using multiple meta−paths can lead to information loss or redundancy, as researchers may not provide all possible meta−paths for large−scale datasets [15]. Taking the common ACM dataset as an example, if PAP and PSP are selected as meta−paths, the target node P can only aggregate 2−hop nodes of the same type P, leading to the loss of information from more distant nodes. Adding longer meta−paths (such as PAPSP) introduces redundant nodes, as shown in Figure 1b (target−type nodes connected by red edges become redundant after adding the PAPSP path). In fact, meta−path−based methods have several shortcomings:

- The definition of meta−paths requires sufficient expertise in the relevant field, and in the case of large−scale datasets, it is almost impossible to enumerate all possible meta−paths [15].
- Models based on meta−paths only utilize extracted meta−path data for training, leading to issues of data loss and redundancy, as illustrated in Figure 1b. Previous studies [13] have demonstrated that the gap between node degrees on the ACM dataset can be as high as 300 times, indicating that the choice of meta−paths significantly influences the performance of network embeddings in various downstream tasks [15].
- Meta−paths constrain the data generalization capability and data−driven adaptive techniques of deep learning. Data generalization capability involves learning the features and patterns of data, abstracting and generalizing the data to extract their essential characteristics [16]. Data−driven adaptive techniques involve training neural networks with large amounts of data to enable them to adaptively learn the features and patterns of the data, thereby enhancing their intelligence and accuracy across various tasks [17]. Meta−paths, by extracting partial data for embedding learning, limit the data generalization capability and data−driven adaptive techniques of deep learning.

Ref. [18] compared HGNNs using meta−paths with GAT that did not use meta−paths and verified that meta−paths did not really play a positive role. Moreover, ref. [13] investigated statistics on the density of different meta−paths and found that there is a huge gap in the degrees of nodes under different meta−paths. This suggests that when predefining meta−paths, there is a high risk of discarding important information. Simultaneously, ref. [18] demonstrated the importance of attention mechanisms. With proper configuration of the GAT model, a simple attention mechanism can outperform many existing HGNNs. Graph convolutional networks (GCNs) have also been validated for their robust feature extraction capabilities [5,7], effectively capturing local structures. As shown in Figure 1a, the GCN, unlike meta−path−based subgraph aggregation, utilizes a "kernel" for feature extraction from all neighbors, addressing the issue of potential information loss due to meta−path selection. However, due to the heterogeneity of nodes, the GCN cannot directly perform feature extraction on heterogeneous graph data.

In summary, to address the limitations of meta−paths and enhance the extraction capability of regional features, this paper proposes the heterogeneous graph neural network model based on regional feature extraction (HGNN−BRFE), which does not rely on meta−paths. The model employs an "extraction−fusion" mechanism to enhance learning capabilities from three aspects: (1) First, in order to avoid the node loss problem caused by meta−paths, we divide the surrounding neighbor nodes into different areas in the preprocessing stage. The homogeneous area can be regarded as a simple homogeneous graph, so a GCN can be used for feature extraction. (2) In order to retain the semantic

information and heterogeneous features of heterogeneous graphs, we extract node features of low−order heterogeneous regions and use attention to fuse them to obtain low−order heterogeneous information. (3) In order to prevent the loss of the node's own features and possible over−smoothing problems, the final node features are composed of high−order homogeneous features, low−order heterogeneous features, and the node's own features.

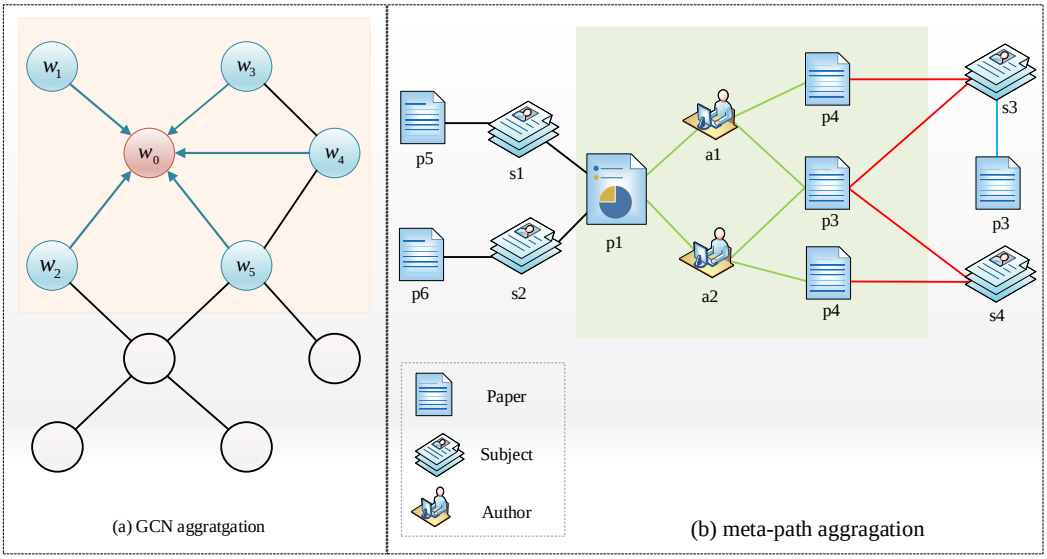


Figure 1. Aggregation methods in GCNs and meta−paths differ. In (**a**), the GCN employs an aggregation method akin to a CNN, aggregating neighboring nodes in a "kernel" manner. In contrast, meta−paths only aggregate the selected nodes, discarding other information. In (**b**), the green box represents the node scope when aggregating with PAP for the target node p1. The nodes connected by red edges are nodes that are repeatedly aggregated after adding PAPSP.

This paper makes the following contributions:

1. We propose a new HGNN−BRFE model that avoids the meta−path aggregation pattern, enabling thorough feature extraction for nodes within designated regions. This approach mitigates the limitations of meta−paths on the data generalization capability and data−driven adaptive techniques of neural networks.
2. The model adopts the "extraction−fusion" strategy to enhance feature extraction capabilities. It divides the dataset into different areas and extracts features by area. At the same time, the different areas are fused in the final node specialties, and the node's own characteristics are retained, effectively preventing over−smoothing.
3. The reliability of the model is validated on three real−world datasets. Experimental results demonstrate the effectiveness of the proposed model, consistently outperforming existing methods across various metrics when compared to state−of−the−art approaches.

2. Related Work

In this section, we describe two main types of HGNNs based on the aggregation of node features: HGNNs based on meta−paths and HGNNs that do not utilize meta−paths.

2.1. HGNNs Based on Meta−Paths

HIN2Vec [12] and HAN [14] are representative heterogeneous graph neural networks that project heterogeneous graphs into homogeneous graphs and apply graph neural networks on them. Taking HAN [14] as an example, it aggregates target nodes selected by predefined meta−paths and uses attention mechanisms for aggregation. It then employs semantic attention for semantic−level aggregation across different meta−paths to obtain the final embedding representation. MAGNN [9], while using meta−paths, thoroughly considers the semantic information of intermediate nodes within meta−paths to extract

richer semantic features. HetGNN [19] conducts random sampling of neighbors near nodes through restartable random walks and aggregates these neighbors within types using Bi−LSTM. GTN [11] calculates meta−path−based graphs for all possible meta−paths within length constraints through iterative multiplication of the adjacency matrices of multi−edge subgraphs, followed by graph convolution operations. CKD [12] introduces a cooperative extraction strategy, modeling knowledge at different granularities within each meta−path and using mutual information for cooperative extraction, explicitly preserving semantics and correlations between meta−paths. HPN [20] theoretically demonstrates that meta−path−based HGNNs and aggregation methods based on random walks are essentially the same, prone to "semantic confusion" after a certain number of aggregations, and proposes a new semantic fusion mechanism to avoid semantic confusion. SeHGNN [21] proposes a simple and efficient heterogeneous graph neural network that precomputes neighbor aggregation in the preprocessing stage, reducing the unnecessary complexity of meta−path−based methods. It also employs a transformer−based aggregation method to further explore the mutual relationships between each pair of semantics. ie−HGCN [10] designs a hierarchical architecture for object and type aggregation, automatically extracting useful meta−paths for each object. RoHE [22] eliminates perturbation amplification and issues with soft attention mechanisms during node−level aggregation by passing probabilities and purifying masks. MHGCN [15], through multi−layer graph convolution modules, can automatically acquire meta−path information across multiple relationships. However, meta−paths suffer from the issue of information loss. HGNN−BRFE performs feature aggregation from different regions, thereby mitigating information loss to some extent.

2.2. HGNNs Without Using Meta−Paths

HGNNs without using meta−paths often simultaneously capture both structural and semantic information during aggregation. These methods employ multi−layer GNN networks, with each layer having attention modules to learn the importance of edge relationships [21]. RSHN [23] integrates the original graph and its coarse−grained graph without the need for any prior knowledge such as meta−paths, embedding nodes and edges into a heterogeneous graph for integration, and updating the central node. HetSAN [24], based on an attention model, aggregates adjacent vertices with different semantics using attention strategies for different types of edges, automatically uncovering semantic information in heterogeneous networks. HGT [25] introduces an attention mechanism related to node and edge types, using meta−relations to parameterize the weight matrix to calculate attention coefficients on each edge. This attention mechanism enables automatic implicit learning and extraction of important information. HGB [18] computes edge type information in the attention network and enhances model capacity through residual connections. However, these methods mostly focus on local connections and have shortcomings in effective data fusion. HGNN−BRFE conducts feature extraction and fusion based on regions, effectively avoiding the issues of graph data disruption and limitations in neural network capabilities caused by discarding nodes, as well as the problem of repeated node selection under meta−paths.

3. Preliminaries

Definition 1 (heterogeneous graph [14]). *A heterogeneous graph can be represented as* $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, *composed of the object set* $\mathcal{V}$ *and edge set* $\mathcal{E}$. *The heterogeneous graph is also associated with object mapping function* $\psi : \mathcal{V} \to \mathcal{A}$ *and edge mapping function* $\varphi : \mathcal{E} \to \mathcal{R}$. $\mathcal{A} = \{\mathcal{A}_i | i \geq 1\}$ *represents the predefined object type set,* $\mathcal{R} = \{\mathcal{R}_i | i \geq 1\}$ *represents the predefined edge type set, and* $|\mathcal{A}| + |\mathcal{R}| > 2$.

Definition 2 (1-layer neighbors). *For any node v in the graph, its 1−layer neighbors are all nodes directly connected to v via a single edge, denoted as* $N(v, 1)$: $N(v, 1) = \{u \mid \exists (v, u) \in \mathcal{E}\}$.

Definition 3 (l-layer neighbors)**.** *For any node v and positive integer l, its l-layer neighbors are all nodes connected to v via l edges, denoted as $N(v,l)$: $N(v,l) = \{u \mid \exists \mu(v,u) = (v,u_1,u_2,\ldots,u_{l-1},u)$ and $((u_i,u_{i+1}) \in \mathcal{E})\}$, where $\mu(v,u)$ indicates the path between v and u.*

Definition 4 (region)**.** *Given a heterogeneous graph $\mathcal{G} = \{\mathcal{V},\mathcal{E}\}$ and a target node $v_i \in \mathcal{V}$, based on the number of layers, we can obtain different sets of nodes at different layers for v_i, denoted as $R = \{R_1,R_2,\ldots,R_L\}$, where $R_l = \{R_l^1,R_l^2,\ldots,R_l^s\}$ represents the set of regions at layer l. For $\forall v_j \in R_l^p$, there is $d(v_i,v_j) = l$ and $\psi(\mu(v_i,v_j))$ is the same. Here, $d(v_i,v_j)$ and $\mu(v_i,v_j)$ represent the distance and path between nodes v_i and v_j, respectively, and $\psi(\mu(v_i,v_j))$ is the function that calculates the node type on the path between v_i and v_j.*

Definition 5 (homogeneous region)**.** *Given a region R_l^p and a node v_i, for $\forall v_j \in R_l^p$, if $\psi(v_i) = \psi(v_j)$, then R_l^p is referred to as a homogeneous region of node v_i.*

Definition 6 (heterogeneous region)**.** *Given a region R_l^p and a node v_i, for $\forall v_j \in R_l^p$, if $\psi(v_i) \neq \psi(v_j)$, then R_l^p is referred to as a heterogeneous region of node v_i.*

Due to the structural characteristics of graphs, a node may simultaneously lie on multiple paths of different lengths, leading to its repeated allocation to different regions, which can cause data redundancy. To avoid this issue, this paper stipulates that each node belongs only to the region at the shortest distance from the target node. Figure 2 provides an example of region partitioning.

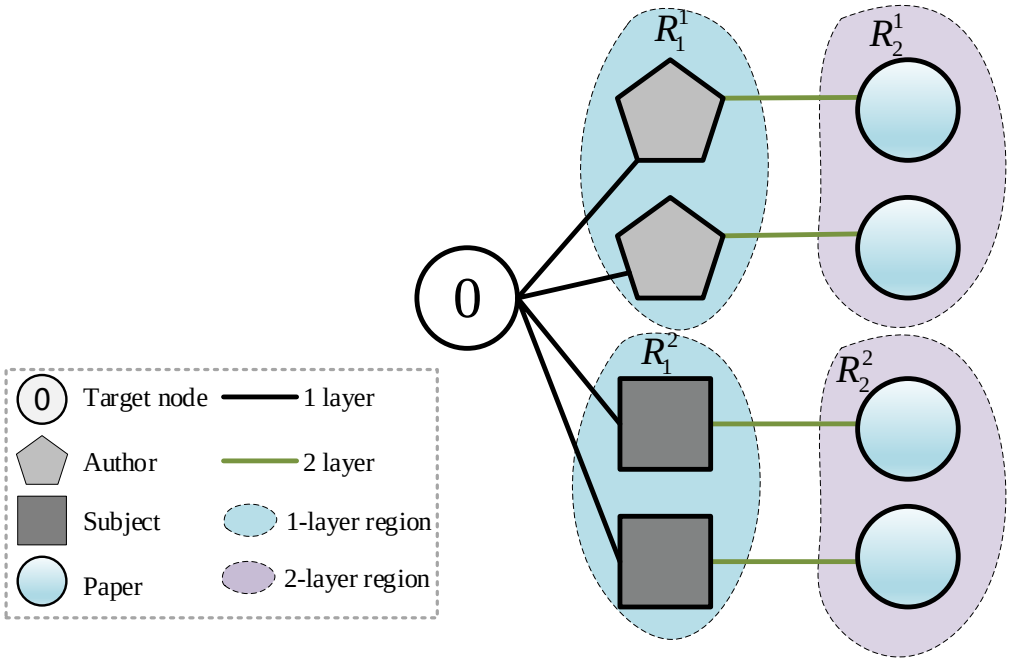


Figure 2. Example of region partitioning.

For heterogeneous regions, the model only extracts features from the nearest region (1−layer neighbors), referred to as low−order heterogeneous region feature extraction; for homogeneous regions, we convert them into a homogeneous subgraph $\mathcal{C}$ and perform feature extraction on regions within the k−layer range, known as high−order homogeneous region feature extraction. Therefore, the l−th layer of graph $\mathcal{C}$ has a set of regions $R_l = \{R_l^1,R_l^2,\ldots,R_l^p\}$.

Problem 1 (heterogeneous graph embedding [26])**.** *Given a heterogeneous graph $\mathcal{G} = \{\mathcal{V},\mathcal{E}\}$, the task of heterogeneous graph embedding is to obtain d-dimensional node representations $h_v \in \mathbb{R}^d$*

for all $v \in \mathcal{V}$, *where* $d \ll |\mathcal{V}|$. *The goal is to capture rich structural and semantic information involved in* $\mathcal{G}$.

4. Model

In this section, we propose a new HGNN−BRFE model. This model aggregates features separately from high−order homogeneous regions and low−order heterogeneous regions, combining the aggregated features with the node's own features to generate the final node embeddings. Since nodes of the same type in a heterogeneous graph are often not directly adjacent, we set the hyperparameter l to aggregate nodes of the same type within l-layer distances.

To extract richer node features, in the preprocessing stage, we divide each layer into different regions to accommodate the feature extraction method of the GCN. Simultaneously, considering that using the same weight matrix for all nodes may lead to an inability to capture "personalized" information for different nodes [10], we extract features from different region information separately and aggregate them using attention, as illustrated in Figure 3. Since nodes of the same type in a heterogeneous graph are often not directly adjacent, we set a hyperparameter k to aggregate nodes of the same type within k orders.

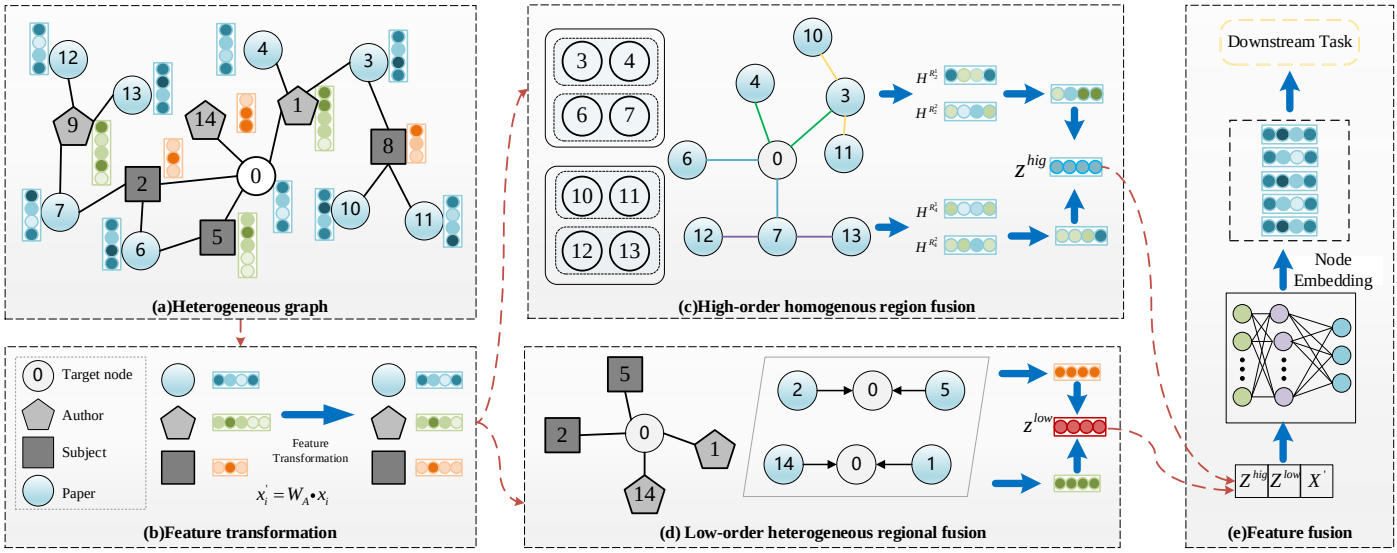


Figure 3. Model framework.

At the same time, to retain heterogeneous information, we employ an attention mechanism to aggregate the heterogeneous neighbors of the target node. Since the heterogeneous neighbors of each node are different, this approch partially alleviates the over−smoothing problem in graph neural networks. In the final node embedding representation, we combine the node's feature representation, high−order homogeneous region features, and low−order heterogeneous region features to obtain a more comprehensive feature representation, which is then applied to downstream tasks.

Due to the heterogeneity of heterogeneous nodes, nodes of different types often reside in different feature spaces. Therefore, for each type of node, we set a type−specific transformation matrix to project it into a common feature space. For a node i of type $\mathcal{A}$, we design a type−specific mapping matrix W_A to transform its feature x_i into the common space.

$$x_i' = W_A \cdot x_i \tag{1}$$

Here, $W_A \in R^{d' \times d}$ is the feature transformation matrix corresponding to type $\mathcal{A}$, where d is the original feature dimension of nodes in class $\mathcal{A}$, and d' is the dimension after projecting features for all types of nodes.

4.1. High−Order Homogeneous Region Aggregation

4.1.1. Region Feature Aggregation

As shown in Figure 3, we partition nodes within each order based on their predecessor node types. For homogeneous regions, we transform them into homogeneous subgraphs $\mathcal{C}$ and perform feature extraction on regions within the k-order range. In the l-hop, we obtain a homogeneous region set $R_l = \left\{R_l^1, R_l^2, \ldots R_l^m\right\}$ with m regions, and we extract features for all nodes within the regions. The feature representation after extracting from region p is then:

$$H^{R_l^p} = \widetilde{R_l^p} X' W_l^p \tag{2}$$

where $\widetilde{R_l^p}$ represents the L2 normalized information of $R_l^p \in R_l$ regions, X' is the node feature after feature transformation, and $H^{R_l^p}$ is the feature representation after aggregating the R_l^p region. The aggregated feature representation of each region in the l-hop can be denoted as $\left\{H^{R_l^1}, H^{R_l^2} \ldots H^{R_l^m}\right\}$.

After obtaining the feature representations of different regions, the model aggregates different regions within the same layer. Due to the varying number of regions in each layer, a simple addition operation would disrupt the distribution of node features and make it difficult to control the composition weight of the final nodes. Similarly, the average operation cannot be employed because it treats each region as equally important, reducing the emphasis on key regions and losing the significance of region−specific feature extraction. Therefore, for different regions within the same layer, we use an attention mechanism for aggregation. This approach preserves the original feature distribution space while allowing features to focus on more important regions.

Through the aggregation of features within the regions, we obtained the feature representations of various regions in the l-hop $\left\{H^{R_l^1}, H^{R_l^2} \ldots H^{R_l^m}\right\}$. Using this as input, we calculate the attention coefficients, as shown below:

$$w_{R_l^p} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} q_s^T \cdot \tanh\left(W_{\text{intra}} \cdot h_i^{R_l^p} + b_{\text{intra}}\right) \tag{3}$$

where W_{intra} is the learnable weight matrix, b_{intra} is the learnable bias vector, and q_s is the semantic−level attention vector. All region embeddings share the above parameters. After obtaining the importance of each region, they are normalized through the softmax function. The weight of region R_l^i can be calculated as $w_{R_l^i}$, as shown below:

$$\alpha_{R_l^p} = \frac{\exp\left(w_{R_l^p}\right)}{\sum_{j=1}^{m} \exp\left(w_{R_l^j}\right)} \tag{4}$$

where $\alpha_{R_l^p}$ can be understood as the contribution of region R_l^p to a specific task. With the weights, the node feature representation for the l-th order can be obtained as:

$$Z_l = \sum_{p=1}^{m} \alpha_{R_l^p} \cdot H^{R_l^p} \tag{5}$$

Thus, we can obtain the collection of aggregated feature information for each order as $Z = \{Z_1, Z_2, \ldots, Z_k\}$.

4.1.2. Inter−Layer Aggregation

In general, nodes that are closer have higher importance. For example, in the ACM dataset, a 2−hop connection may represent two papers written by the same author, while a 4−hop connection implies that two papers share a common author who has written another paper. However, valuable information also exists in distant nodes. Therefore, we

use an attention mechanism to let the model learn the weights of different orders, allowing better aggregation of node features at each layer.

We employ an attention mechanism to aggregate node features at each layer. The aggregated node feature information from different layers, denoted as $Z = \{Z_1, Z_2, \ldots, Z_k\}$, is used as input, and its weight coefficients and attention are calculated through a neural network layer as follows:

$$w_{R_l} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} q_z^T \cdot \tanh\left(W_{hig} \cdot z_l + b_{hig}\right) \tag{6}$$

$$\beta_l = \frac{\exp\left(w_{R_l}\right)}{\sum_{j=1}^{k} \exp\left(w_{R_j}\right)} \tag{7}$$

With the weights, we can obtain the node feature representation after high−order homogeneous feature extraction:

$$Z^{hig} = \sum_{l=1}^{k} \beta_l \cdot Z_l \tag{8}$$

4.2. Low−Order Heterogeneous Region Extraction

After extracting features for all homogeneous nodes within the k−order range, we extract features of heterogeneous neighboring nodes of the target node in low−order heterogeneous regions. This has the following advantages: (1) Low−order heterogeneous regions are directly connected to nodes, and directly connected heterogeneous nodes have a close relationship with the target node; (2) rich information in the heterogeneous graph is fully extracted; (3) the low−order heterogeneous neighbors of each node are different, which to some extent alleviates the feature over−smoothing problem.

First, assuming that there are $\Phi = \{\Phi_1, \Phi_2, \ldots, \Phi_S\}$ other types of nodes directly adjacent to the target node, we first apply node−level attention to fuse neighbors of type Φ_m:

$$h_i^{\Phi_m} = \sigma\left(\sum_{j \in N_i^{\Phi_m}} \alpha_{i,j}^{\Phi_m} \cdot h'_j\right) \tag{9}$$

where σ is the activation function, h'_j is the feature−transformed node feature for node j, and $\alpha_{i,j}^{\Phi_m}$ is the attention value for node j of type Φ_m, calculated as follows:

$$\alpha_{i,j}^{\Phi_m} = \frac{\exp\left(LeakyReLU\left(q_{\Phi_m}^{\top} \cdot \left[h'_i || h'_j\right]\right)\right)}{\sum_{l \in N_i^{\Phi_m}} \exp\left(LeakyReLU\left(q_{\Phi_m}^{\top} \cdot \left[h'_i \| h'_l\right]\right)\right)} \tag{10}$$

where $\|$ represents the concatenation operation, and $q_{\Phi_m}^{\top}$ is the attention vector for the node layer, used to compute the attention values.

After obtaining embeddings for all heterogeneous node types $[h_i^{\Phi_1}, h_i^{\Phi_2}, \ldots, h_i^{\Phi_s}]$, we use an attention mechanism to aggregate different heterogeneous information, ensuring that the most relevant heterogeneous nodes contribute more to the final representation.

$$z_i^{low} = \sum_{m=1}^{S} \gamma_{\Phi_m} \cdot h_i^{\Phi_m} \tag{11}$$

where γ_{Φ_m} is the calculated attention value for the neighbors of type Φ_m, computed similarly to Formula (7), and z_i^{low} is the feature representation of node i after aggregating low−order heterogeneous information.

4.3. Model Training

Through the above steps, we obtained high−order homogeneous embeddings and low−order heterogeneous embeddings for nodes. These embeddings are concatenated with the initial features of the nodes, and then, through a neural network layer, we obtain the final embedding representation for the nodes. Since we used attention mechanisms in previous modules for intra−order, inter−order, and inter−type aggregation, the node features of each part are on the same "scale", making this operation fair for all parts of the features.

$$z_i = W_{fin} \cdot \left[z_i^{low} \middle\| z_i^{hig} \middle\| h_i' \right] + b_{fin} \tag{12}$$

where W_{fin} and b_{fin} are learnable parameters, with $W_{fin} \in R^{3d \times d}$. With this, we obtain the final embedding representation for the nodes for downstream tasks.

After obtaining the final embedding for the nodes, we optimize the entire model by minimizing the following loss:

$$\mathcal{L} = -\sum_{l \in Y_L} Y_{v_l} \cdot \ln\left(C \cdot z_{v_l}\right) \tag{13}$$

where Y_{v_l} and z_{v_l} are the label and embedding vector for node v_l, respectively, and C represents the classifier parameters. The overall learning algorithm is outlined in Algorithm 1.

Algorithm 1 The overall learning algorithm of HGNN−BRFE

Input: the heterogeneous graph $G = \{V, \mathcal{E}\}$, the initial node feature X, region set R, heterogeneous neighbor set Φ, the HGNN−BRFE model.
Output: The node Embeddings Z.

for node type $A_i \in A$ **do**
  Node feature transformation by Equation (1)
end for
* *feature extraction for regions within the k−layer range* * \
for $l = 1 \ldots k$ **do**
  for region $R_l^t \in R_l$ **do**
    Extract regional feature $H^{R_l^t}$ by Equation (2)
  end for
  for $H^{R_l^i} \in H^{R_l}$ **do**
    Calculate the region importance of $\alpha_{R_l^i}$ by Equations (3) and (4)
  end for
end for
* *feature extraction for heterogeneous regions* * \
for Φ_m in Φ **do**
  Extracting heterogeneous type features H^{Φ_m} by Equations (9) and (10)
end for
Compute high−order homogeneous feature Z^{high} by Equation (8)
Compute low−order heterogeneous feature Z^{low} by Equation (11)
Calculate the final embedding Z by Equation (12)
Backpropagation and update parameters according to Equation (13)

5. Experiment

In this section, we conducted numerous experiments, including node classification, node clustering, ablation study, and parameter sensitivity analysis, to demonstrate the performance of HGNN−BRFE on real−world datasets.

5.1. Datasets

We evaluated the effectiveness of the model on three real−world heterogeneous graph datasets. The statistical information of the datasets is presented in Table 1.

Table 1. Datasets information.

Dataset	Target Node	Training	Validation	Test	Node Type	Link Type
ACM	4019	400	400	3219	Paper (P) Author (A) Subject (S)	P-P P-A P-S
DBLP	4057	400	400	3257	Author (A) Paper (P) Term (T) Venue (V)	A-P P-T P-V
IMDB	4278	400	400	3878	Movie (M) Director (D) Actor (A)	M-D M-A

ACM [8]: A subset of the ACM dataset, consisting of 4019 papers (P), 7167 authors (A), and 60 subjects (S).

DBLP [9]: A subset of the DBLP dataset, comprising 14,328 papers (P), 8789 terms (T), 4057 authors (A), and 20 venues (V).

IMDB [9]: A subset of the IMDB dataset, including 4278 movies (M), 5841 actors (A), and 2269 directors (D).

5.2. Baselines

We will compare the proposed HGNN−BRFE with nine baselines:

R−GCN [27] (2018): The relationship between nodes is not considered in the GCN, and R−GCN solves the problem of using a GCN to deal with the influence of different edge relations on nodes in graph structure.

HAN [14] (2019): It transforms a heterogeneous graph into multiple homogeneous graphs through meta−paths and integrates node representations under different semantics using semantic attention.

HetGNN [19] (2019): It captures interactions between strongly correlated heterogeneous neighbors through neighbor sampling and aggregates node information using Bi−LSTM.

MAGNN [9] (2020): It learns heterogeneous graph representations through intra−path and inter−path aggregations. For simplicity, we refer to it as MAGN.

GTN [11] (2019): It multiplies the adjacency matrices of multiple subgraphs to obtain neighborhood graphs of different meta−paths, automatically discovering valuable meta−paths.

HGSL [28] (2021): It reconstructs a more information−rich heterogeneous graph by integrating multiple information graphs.

ie−HGCN [10] (2023): It aids in meta−path discovery in HIN through a specific type−level attention mechanism. For simplicity, we refer to it as ieHG.

RoHe [22] (2022): It introduces meta−path−based transition probabilities as priors for purifiers, aggregates neighbor information for all meta−paths using purified attention, semantically aggregates these meta−paths, and generates node embeddings for downstream tasks.

AC−HEN [29] (2022): AC−HEN consists of feature aggregation, structural aggregation, and multi−view embedding fusion modules. It aggregates features of the target node by weighted aggregation of its k−nearest neighbors (kNNs), learns node embeddings using structural aggregation considering both first−order and high−order network structures, and distinguishes different contributions of adjacent nodes through weighted aggregation.

5.3. Experimental Setup

For a fair comparison, we set the embedding dimensions the same for all methods ($d = 64$). For models requiring meta−paths [9,14,22], we used PAP, PSP, and PAPSP for the ACM dataset, APA, APTPA, and APVPA for DBLP, and MAM, MDM, and MAMDM for IMDB. For models employing different feature processing methods [10], we processed the initial features through linear transformations and used them as inputs for the models. For

models with attention mechanisms [9,14,22], the number of attention heads K was set to 8. For the aggregation distance of HGNN−BRFE, we aggregate all homogeneous regions within $l = 4$. Different values of l will be studied in the hyperparameter section, Section 5.8.

5.4. Node Classification

We evaluated the models on a multi−classification task, using Macro−F1 (Ma−F1) and Micro−F1 (Mi−F1) as evaluation metrics. The results are presented in Table 2, where the best results are highlighted in bold.

Table 2. Classification results of different models on three datasets (%).

Datasets	Metrics	Ratio	R-GCN	HAN	HetG	MAG	GTN	HGSL	ieHG	RoHE	AC-HEN	HGNN-BRFE
ACM	Ma-F1	0.4	86.32	91.33	90.13	89.6	91.35	92.58	92.13	91.99	91.12	**93.58**
		0.6	87.16	91.73	90.72	90.48	91.73	92.73	92.58	92.20	91.62	**93.93**
		0.8	87.52	91.91	91.01	90.89	91.82	92.83	92.78	92.43	92.11	**94.22**
	Mi-F1	0.4	86.60	91.22	90.11	89.70	91.23	92.54	92.12	91.92	91.19	**93.48**
		0.6	87.37	91.60	90.63	90.51	91.62	92.69	92.54	92.10	91.71	**93.84**
		0.8	87.73	91.76	90.94	90.91	91.71	92.77	92.74	92.34	92.16	**94.10**
DBLP	Ma-F1	0.4	75.36	92.87	92.17	93.32	94.26	93.64	93.56	92.77	93.24	**94.11**
		0.6	75.78	93.05	92.52	93.69	94.16	93.80	93.65	92.84	93.69	**94.21**
		0.8	76.11	93.16	92.86	94.01	94.26	94.09	94.09	93.11	93.81	**94.38**
	Mi-F1	0.4	76.42	93.43	92.65	93.82	**94.71**	94.08	94.01	93.28	93.74	94.54
		0.6	76.77	93.61	92.97	94.18	94.60	94.24	94.11	93.34	94.19	**94.64**
		0.8	77.18	93.69	93.05	94.48	94.70	94.51	94.46	93.55	94.29	**94.78**
IMDB	Ma-F1	0.4	49.98	58.56	51.61	59.23	57.90	58.23	59.34	58.37	59.71	**62.41**
		0.6	50.61	58.73	53.22	59.72	58.03	58.81	59.64	58.63	59.83	**62.60**
		0.8	51.03	58.88	53.23	59.94	58.83	58.94	59.86	58.55	59.78	**62.59**
	Mi-F1	0.4	49.99	58.58	52.85	59.29	57.82	58.57	59.25	58.63	59.45	**62.47**
		0.6	50.55	58.72	53.91	59.80	57.88	58.97	59.56	58.87	59.97	**62.68**
		0.8	51.01	58.91	54.37	60.06	58.73	59.09	58.83	58.78	60.03	**62.69**

As shown in Table 2, compared with HAN, HGNN−BRFE increased by 2.31% and 2.34% in the ACM dataset, 3.71% and 3.78% in the IMDB dataset, and 1.22% and 1.09% in the DBLP dataset, respectively. Compared with MAGNN, the results showed an increase of 3.33% and 3.19% on the ACM dataset, 2.65% and 2.63% on the IMDB dataset, and 0.37% and 0.30% on the DBLP dataset, respectively. The main reason is that our model avoids the deficiency of meta−paths and further improves the feature extraction ability. Compared with RoHE, it has increased by 1.79% and 1.76% points on the ACM dataset, 3.91% and 4.04% on the IMDB dataset, and 1.27% and 1.23% on the DBLP dataset, respectively. The main reason is that the neighbor aggregation method of RoHE is still based on the meta−path, but the region aggregation method in this paper can avoid the redundancy and information loss of the meta−path.

We believe that the main reason for the improvement on the ACM and IMDB datasets is that the aggregation method by region can aggregate more abundant feature representations and enhance the expression ability of the model, and the slight improvement on the DBLP dataset is primarily constrained by dataset characteristics. The divided region exhibits similarity to the meta−path data used in other papers, resulting in a relatively minor enhancement effect. Nevertheless, this observation also serves to underscore the positive significance of regional aggregation.

5.5. Node Clustering

The learned node embeddings are measured using average Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI), and clustering is performed using K−means. The results are shown in Table 3, where the best results are highlighted in bold.

Table 3. The results of clustering experiments for different models (%).

Datasets	ACM		DBLP		IMDB	
	NMI	**ARI**	**NMI**	**ARI**	**NMI**	**ARI**
MAGNN	0.7016	0.7214	0.7866	0.8401	0.1309	0.1277
ie-HGCN	0.4947	0.3489	0.3234	0.2722	0.1308	0.1305
HGSL	0.7025	0.7425	0.7762	0.8248	0.0625	0.0877
RoHe	0.6921	0.7248	0.7083	0.7727	0.1238	0.1287
AC-HEN	0.7045	0.7388	0.7718	0.8204	0.0933	0.1011
HGNN-BRFE	**0.7411**	**0.7892**	**0.7995**	**0.8558**	**0.1321**	**0.1377**

The model showed an improvement of 3.95% and 6.78% compared to MAGNN on the ACM dataset. When compared with ie−HGCN, the improvement was 24.62% and 44.03%, and compared to RoHe, it increased by 4.9% and 6.44%, respectively. The results indicate that HGNN−BRFE has a stronger feature acquisition capability, contributing to enhanced clustering performance.

5.6. Visualization

To provide a more intuitive evaluation, we use t−SNE [30] to project node embeddings into a two−dimensional space and color them based on their labels. Figure 4 illustrates the visualization results of all methods on the ACM dataset. Compared to other models, our model exhibits larger gaps between node types, indicating a more distinct separation of node types.

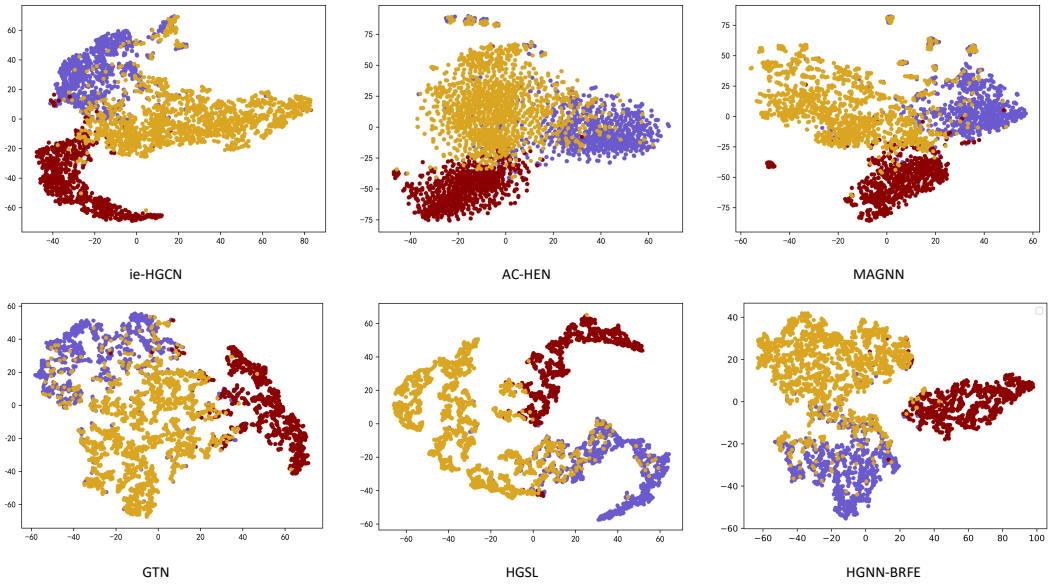


Figure 4. Visualization results of different models.

5.7. Ablation Study

We designed the following variants for ablation studies, using the ACM dataset to evaluate the effects of different aggregation methods.

BRFE_onehot: this method only considers the features of papers as input, replacing features of other types of nodes with one−hot vectors.

To validate the effects of different modules in the model, we individually removed the low−order heterogeneous aggregation, high−order homogeneous aggregation, and self−features of the model, comparing their experimental results in node classification. Specifically:

BRFE$_{-w/o-low}$: discards low−order heterogeneous information in the model to assess the importance of low−order heterogeneous features.

BRFE$_{-w/o-hig}$: aggregates only low−order heterogeneous features, excluding high−order homogeneous node information.

BRFE$_{-w/o-self}$: does not use the original features of nodes in the final node feature embedding.

Figure 5 shows the results of all the above variants on the ACM dataset. From the experimental results, it can be found that compared with other variants, BRFE_onehot and BRFE$_{-w/o-low}$ have higher accuracy, but both are lower than HGNN−BRFE. Both variants reduce the role of low−order heterogeneous information in the model, indicating that the optimization effect of low−order heterogeneous information on the model is lower on the ACM dataset. Similarly, this can also be seen from BRFE$_{w/o-hig}$, which achieved the lowest accuracy, which was further confirming the observation. After removing high−order homogeneous region features, the model achieved the lowest accuracy, indicating that nodes of the same type can obtain more meaningful information. Compared with HGNN−BRFE, the accuracy of BRFE$_{-w/o-self}$ has dropped significantly, indicating that retaining the characteristics of the target node itself is of great help in improving the accuracy of the model. Comprehensive comparison shows that the experimental results of these variants have declined, indicating that each module of our model plays a positive role to a certain extent.

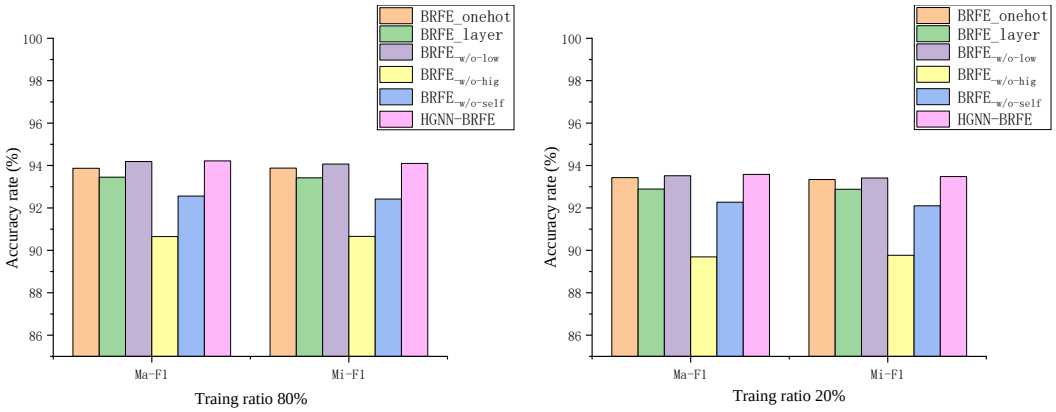


Figure 5. Task performance of HGNN−BRFE and its variants on node classification.

5.8. Parameter Sensitivity Analysis

We investigated the impact of the aggregation range k on the model and reported the performance of node classification and clustering tasks on three datasets with different parameter settings. In this paper, k is set to aggregate regions within three different distances: [2, 4, 6]. This setting is based on the following two considerations: (1) In the three public datasets, homogeneous regions are not directly connected to the target node, so the connection model of HIN needs to be followed; (2) the correlation between nodes at greater distances is lower, but the number of nodes increases exponentially, significantly increasing the training cost. Therefore, k is set to [2, 4, 6]. In fact, when $k = 6$, the distance already far exceeds the existing meta−paths.

From the results shown in Figure 6, we can observe that when $k = 4$, both the Macro−F1 and Micro−F1 metrics for classification results, as well as the NMI and ARI metrics for clustering results, achieved optimal performance on both the ACM and DBLP datasets. For the IMDB dataset, the NMI metric for clustering slightly decreased compared to $k = 2$ and $k = 6$, but overall, the results indicate that the model performs optimally when $k = 4$. When $k = 6$, as the number of layers increases, the aggregation range of nodes is too large, and distant nodes have smaller correlations, thus generating noise and reducing the accuracy of the model.

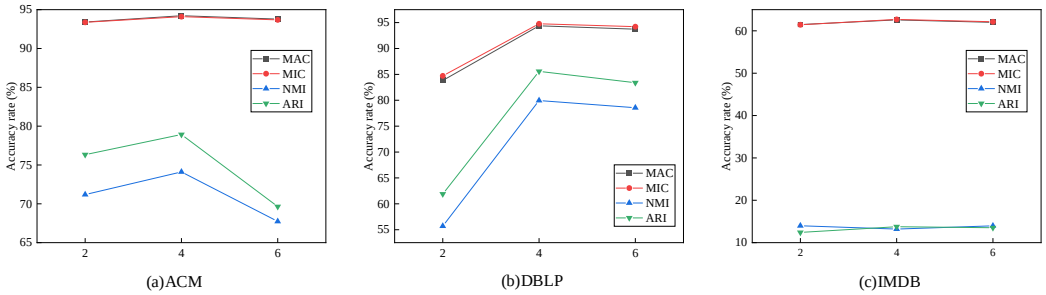


Figure 6. Task performance of HGNN−BRFE and its variants on node classification.

5.9. Attention Study

To further understand the model's performance, we investigated the attention values between different regions, layers, and high−order and low−order features. The results are shown in Table 4, and we found that, as previously mentioned, the closer the distance, the higher the importance. It can be seen that, on the ACM dataset, the attention value for layer 1 (2−hop) is 0.9607, while the attention value for layer 2 (4−hop) is only 0.0393. Within layer 1, the importance of region 1 is 0.99919, while the importance of region 2 is only 0.00081. Regions are divided based on the predecessor nodes of the target type nodes, indicating that nodes between the same authors are more important to each other on the ACM dataset. The experimental results on DBLP are similar to the ACM dataset, and we can draw the same conclusion. However, the attention value for layer 2 in DBLP is much larger than ACM, indicating that features in 4−hop have a positive impact on improving the node embedding level. Similarly, the experimental results on IMDB further validate this point, suggesting that high−order information has a positive impact on enhancing the model's effectiveness.

Regarding the extraction of low−order heterogeneous information, since DBLP has only one direct heterogeneous neighbor, its attention is 1. On the other hand, ACM and IMDB focus on one type of heterogeneous neighbor. Specifically, in the ACM dataset, the influence of Author (A) on Paper (P) is more significant, while in the IMDB dataset, the influence of Director (D) on Movie (M) is more pronounced.

Table 4. Attention values between different regions and different layers.

		ACM	**DBLP**	**IMDB**
Higher order	layer1 regions	0.99919 0.00081	0.1064 0.8936	0.5078 0.4922
	layer2 regions	0.0756 0.0020 0.9224 -	0.1048 0.6274 0.2679 -	0.2206 0.2895 0.229 0.2608
	interlayer attention	0.9607 0.0393	0.7289 0.2711	0.584 0.416
Lower order	type1 type2	0.99996 0.00004	1 -	0.9616 0.0384

5.10. Research on Node Feature Similarity

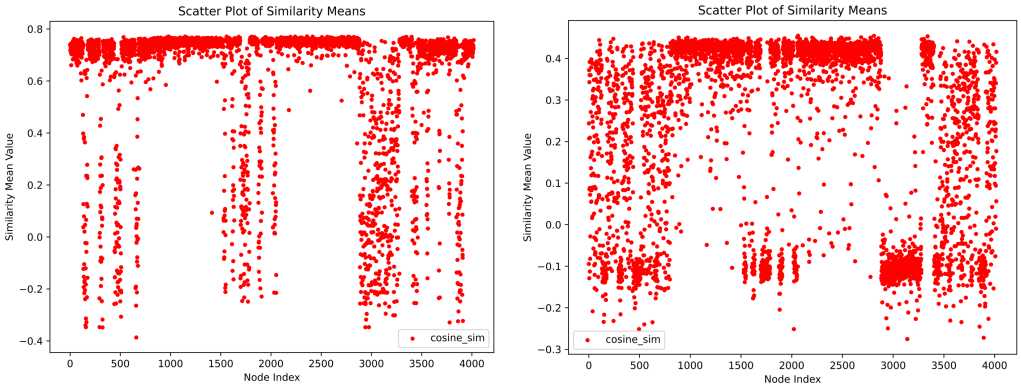
Node feature similarity is a crucial criterion for assessing various characteristics of nodes, such as the smoothing effect. In this subsection, we will compare HGNN−BRFE and HAN on the ACM dataset using cosine similarity and Euclidean distance similarity as evaluation metrics. We will examine the transformations of these two metrics both before and after model execution.

Cosine Similarity: Cosine similarity measures the cosine of the angle between two vectors, with values ranging from −1 to 1. A value closer to 1 indicates greater similarity

between two vectors, while a value closer to −1 indicates greater dissimilarity. When two vectors are entirely different (i.e., the angle between them is 90 degrees), their cosine similarity is 0.

To facilitate the observation of the distribution of similarity characteristics, we calculate the average similarity of each node with other nodes, obtaining $cosine_avg \in R^{4019}$. Each element in the matrix represents the average similarity of the node with other nodes.

We present the average similarity values in the form of a scatter plot, as shown in Figure 7.



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

Figure 7. Scatter plot of the mean cosine similarity embedded by HAN and HGNN−BRFE nodes. (**a**) HAN; (**b**) HGNN−BRFE.

From the graph, it can be observed that the cosine similarity of node features aggregated by HAN is predominantly concentrated around 0.8, while the cosine similarity of node features aggregated by HGNN−BRFE is mostly centered around 0.4. Moreover, there is a considerable distribution near −0.1 for HGNN−BRFE, indicating lower cosine similarity of nodes after aggregation by HGNN−BRFE.

6. Conclusions

This paper first analyzes the issues associated with path−based heterogeneous graph neural networks and proposes the HGNN−BRFE model based on these observations. The HGNN−BRFE model conducts feature extraction for homogeneous nodes based on regions. To prevent over−smoothing and simultaneously retain heterogeneous information in the graph, we designed a low−order heterogeneous feature aggregation module. Furthermore, we preserve the original information of nodes in the final node embedding. Experimental results demonstrate that the HGNN−BRFE model exhibits satisfactory performance in comparison to existing models in both node classification and node clustering tasks.

Despite the strong performance demonstrated by HGNN−BRFE, there are still several areas that require improvement. Firstly, the current model primarily focuses on static heterogeneous graphs, and how to effectively apply it to dynamically changing heterogeneous graph structures requires further research. Secondly, although the attention mechanism helps improve model performance, its computational cost is relatively high, especially when dealing with large−scale datasets, which could become a bottleneck in practical applications. Lastly, while the model performs well across multiple tasks, its adaptability and robustness in certain specific application scenarios still need further validation. Future work will revolve around these issues, including exploring more efficient computational methods, optimizing the model to accommodate a wider variety of heterogeneous graph data, and enhancing the model's performance in complex environments.

Author Contributions: Y.Z.: Conceptualization, Methodology, Software, Writing—original draft. H.D.: Formal analysis, Funding acquisition, Supervision, Writing—review and editing. S.X.: Data curation, Investigation, Visualization. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key R&D Program of China [2022ZD0119501]; NSFC [52374221]; Sci. & Tech. Development Fund of Shandong Province of China [ZR2022MF288, ZR2023MF097]; the Taishan Scholar Program of Shandong Province [ts20190936].

Data Availability Statement: The original data presented in the study are openly available in DBLP at https://dblp.uni-trier.de/ (accessed on 14 October 2024).

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

References

1. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017.
2. Wang, D.; Cui, P.; Zhu, W. Structural deep network embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1225–1234.
3. Fout, A.; Byrd, J.; Shariat, B.; Ben-Hur, A. Protein interface prediction using graph convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017.
4. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926.
5. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
6. Zhang, J.; Shi, X.; Zhao, S.; King, I. Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems. *arXiv* **2019**, arXiv:1905.13129.
7. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv* **2019**, arXiv:1902.10197.
8. Wang, X.; Liu, N.; Han, H.; Shi, C. Self-supervised heterogeneous graph neural network with co-contrastive learning. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 1726–1736.
9. Fu, X.; Zhang, J.; Meng, Z.; King, I. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 2331–2341.
10. Yang, Y.; Guan, Z.; Li, J.; Zhao, W.; Cui, J.; Wang, Q. Interpretable and Efficient Heterogeneous Graph Convolutional Network. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 1637–1650. [CrossRef]
11. Yun, S.; Jeong, M.; Kim, R.; Kang, J.; Kim, H.J. Graph transformer networks. In Proceedings of the Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019.
12. Fu, T.y.; Lee, W.C.; Lei, Z. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1797–1806.
13. Wang, C.; Zhou, S.; Yu, K.; Chen, D.; Li, B.; Feng, Y.; Chen, C. Collaborative knowledge distillation for heterogeneous information network embedding. In Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 1631–1639.
14. Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; Yu, P.S. Heterogeneous graph attention network. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2022–2032.
15. Yu, P.; Fu, C.; Yu, Y.; Huang, C.; Zhao, Z.; Dong, J. Multiplex heterogeneous graph convolutional network. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 2377–2387.
16. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM* **2021**, *64*, 107–115. [CrossRef]
17. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef] [PubMed]
18. Lv, Q.; Ding, M.; Liu, Q.; Chen, Y.; Feng, W.; He, S.; Zhou, C.; Jiang, J.; Dong, Y.; Tang, J. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 1150–1160.
19. Zhang, C.; 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; pp. 793–803.
20. Ji, H.; Wang, X.; Shi, C.; Wang, B.; Philip, S.Y. Heterogeneous graph propagation network. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 521–532. [CrossRef]

21. Yang, X.; Yan, M.; Pan, S.; Ye, X.; Fan, D. Simple and efficient heterogeneous graph neural network. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 10816–10824.
22. Zhang, M.; Wang, X.; Zhu, M.; Shi, C.; Zhang, Z.; Zhou, J. Robust heterogeneous graph neural networks against adversarial attacks. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36, pp. 4363–4370.
23. Zhu, S.; Zhou, C.; Pan, S.; Zhu, X.; Wang, B. Relation structure-aware heterogeneous graph neural network. In Proceedings of the 2019 IEEE international conference on data mining (ICDM), Beijing, China, 8–11 November 2019; pp. 1534–1539.
24. Hong, H.; Guo, H.; Lin, Y.; Yang, X.; Li, Z.; Ye, J. An attention-based graph neural network for heterogeneous structural learning. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 4132–4139.
25. Hu, Z.; Dong, Y.; Wang, K.; Sun, Y. Heterogeneous graph transformer. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 2704–2710.
26. Jin, D.; Huo, C.; Liang, C.; Yang, L. Heterogeneous graph neural network via attribute completion. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 391–400.
27. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, 3–7 June 2018; pp. 593–607.
28. Zhao, J.; Wang, X.; Shi, C.; Hu, B.; Song, G.; Ye, Y. Heterogeneous graph structure learning for graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 4697–4705.
29. Wang, K.; Yu, Y.; Huang, C.; Zhao, Z.; Dong, J. Heterogeneous graph neural network for attribute completion. *Knowl.-Based Syst.* **2022**, *251*, 109171. [CrossRef]
30. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.