GWNN-HF : Beyond assortativity in Graph Wavelet Neural Network

BinFeng Huang
Anhui University
WenJie Zheng
Nanjing University of Science and Technology
Fulan Qian (✉ qianfulan@hotmail.com)
Anhui University
Shu Zhao
Anhui University
Jie Chen
Anhui University
Yanping Zhang
Anhui University

Research Article

Keywords: Graph neural network, Graph wavelet neural network,Assortative networks, Disassortative networks

Posted Date: July 22nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1854505/v1

GWNN-HF : Beyond assortativity in Graph Wavelet Neural Network

BinFeng Huang[1,2,3], WenJie Zheng[4], Fulan Qian[1,2,3*], Shu Zhao[1,2,3], Jie Chen[1,2,3] and Yanping Zhang[1,2,3]

[1]Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Anhui University, Hefei, Anhui Province 230039, China.

[2]School of Computer Science and Technology, Anhui University, Hefei 230601, China.

[3]Information Materials and Intelligent Sensing Laboratory of Anhui Province.

[4]School of Computer Science and Engineering, Nanjing University of Science and Technology, China.

*Corresponding author(s). E-mail(s): qianfulan@hotmail.com; Contributing authors: huangzhanglao123@163.com; wjzheng1996@gmail.com; zhaoshuzs2002@hotmail.com; chenjie200398@163.com; zhangyp2@gmail.com;

Abstract

Graph wavelet neural network exerts a powerful learning ability in the assortative network where most of adjacent nodes have the same label as the target node. However, it doesn't perform well in the disassortative network where most of adjacent nodes have different label than the target node. So graph wavelet neural network cannot extract the most useful information based on different types of networks. On the one hand, as a low-pass filter, graph wavelet neural network only obtains the commonality of the same label nodes, it is not capable of obtaining the difference of different label nodes. On the other hand, graph wavelet neural network only aggregates neighbor nodes so that it can't obtain information of nodes which have similar feature with the target node and are far from the target node. To solve the above problems, we propose the GWNN-HF model, which can effectively adapt to different types of networks and get a better node representation. Specifically speaking, firstly, we design low-pass filter and high-pass filter convolution kernels to get low-pass and high-pass signals and then use adaptive fusion method to fuse them, which effectively get commonality of same label nodes and difference of different label nodes. Secondly, we use the Relaxed Minimum-Spanning Tree algorithm to construct a feature correlation graph and use an attention mechanism to fuse the original graph and feature correlation graph representation. Extensive experiments on benchmark datasets clearly indicates that GWNN-HF behaves better than the state-of-the-art GNNs.

Keywords: Graph neural network, Graph wavelet neural network,Assortative networks, Disassortative networks

1 Introduction

Networks can be seen everywhere in real life, including assortative networks and disassortative networks. The assortative networks are ubiquitous, such as citation networks and social networks[1] [2]. In assortative networks, most of adjacent nodes have the same label as the target node[3]. There can also be many disassortative networks, such as chemical interactions in proteins often occur between different types of amino acids. In disassortative networks, most of adjacent nodes have different labels than the target node[4–7].

In recent years, graph neural networks have attracted extensive attention because of their excellent performance in graph representation learning. It can embed graph into a low-dimensional space by aggregating and transforming adjacent node's information[8–11]. Graph wavelet neural network is called one of the most attractive graph neural network models, which tries to address the limitations of GCN by using graph wavelet transform as the theoretical basis instead of graph Fourier transform[12]. First of all, traditional GCNs employ the Fourier transform as their theoretical basis generally have high computational costs caused by eigendecom-position and frequent multiplications between dense matrices, graph wavelet transform effectively avoid this problem, which has a lower computational cost and more sparse than Fourier transform. Once again, wavelets in GWNN are highly localized in the vertex domain, such localized property can make GWNN more flexible to adjust the receptive fields of nodes (via the scaling operation)[13]. So the two advantages improve the semi-supervised node classification task and learns a better node representation.

Nowadays, the lately published Graph neural networks(including GWNN) can learn a good node representation in the assortative network[14–17]. However, graph neural networks face the problem that ability of learning node representation in disassortative networks declines rapidly. In order to improve graph neural networks can extract the most useful information based on different types of networks, many GCN-based models have been proposed to improve graph graph neural networks ability of learning node representation in different kinds of networks. For example, FAGCN [18] designs a novel graph convolutional network to adaptively combine the low-pass and high-pass signals, but it only cares about the information of the neighbor node. Geom-GCN [19] utilizes the structural similarity to capture the long-range dependencies in the disassortative network, but it fails to obtain difference of different label nodes.

The above-mentioned GCN-based models can be well adapted to different types of networks. However, no one has proposed an effective method to make graph wavelet neural network learn a better node representation regardless of the type of network in recent years. The problem is currently mainly faced with the following challenges. On the one hand, how to design a new filter to make GWNN effectively adapt to different types of networks. On the other hand, how to make full use of the role of feature correlation to assist in improving node representation.

In order to handle challenges of graph wavelet neural network in learning node representation. Here, we propose the GWNN-HF model. Specifically, as shown in Figure 1. Firstly, Based on graph wavelet neural network, we design low-pass and high-pass filters to get corresponding low-pass signals and high-pass signals which can shorten or enlarge the distance automatically between nodes in the original graph and feature correlation graph. Secondly, we build a feature correlation graph based on node feature to make nodes with similar feature connect each other, then we use attention mechanism to make the node representations which are from feature correlation graph and original graph combine adaptatively. The method can make the target node adaptively obtain node information with similar feature, so that the target node can learn a better representation. Our contributions can be summarized as follows:

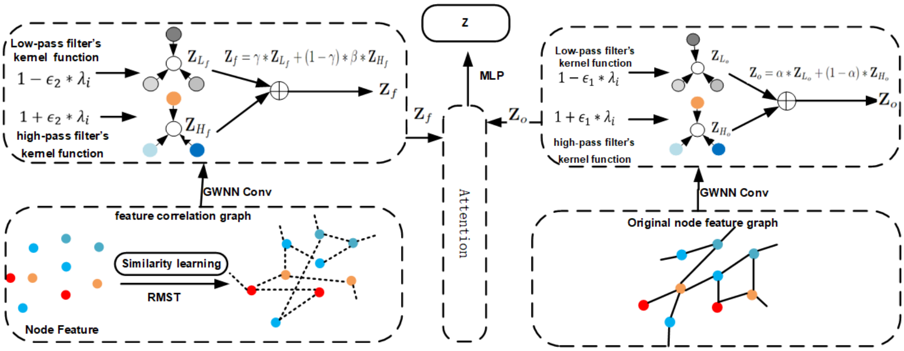
Low-pass filter's kernel function
$1-\epsilon_2*\lambda_i$
$1+\epsilon_2*\lambda_i$
high-pass filter's kernel function
Z_{L_f}
Z_{H_f}
$Z_f=\gamma*Z_{L_f}+(1-\gamma)*\beta*Z_{H_f}$
Z_f
Z
MLP
Z_f
Z_o
Attention
Low-pass filter's kernel function
$1-\epsilon_1*\lambda_i$
$1+\epsilon_1*\lambda_i$
high-pass filter's kernel function
Z_{L_o}
Z_{H_o}
$Z_o=\alpha*Z_{L_o}+(1-\alpha)*Z_{H_o}$
Z_o
GWNN Conv
feature correlation graph
Similarity learning
RMST
Node Feature
GWNN Conv
Original node feature graph

Fig. 1 The overall framework of the proposed GWNN-HF. The bottom left part represents the processing flow of the feature correlation graph, in which the dotted line represents the edge generated according to the feature similarity. The bottom right part represents the processing flow of the original feature graph, in which the solid line represents the edge that exists in the original feature structure.

- We propose GWNN-HF that designs two innovative filters that efficiently integrate low-pass and high-pass signals for original graph and feature correlation graph respectively to enhance adaptability in assortative networks and disassortative networks.
- To take full advantage of feature correlation, we use Relaxed Minimum Spanning Tree algorithm to construct feature correlation graph which can capture the global information from distant nodes with similar feature in the original graph.
- Experiments show that the proposed model is superior to the representative graph neural network methods.

2 Related work

2.1 Graph neural network

From the perspective of spatial domain, graph neural networks usually focus on aggregating and transforming neighborhood information with different designs. GraphSage [8] takes first-order neighbors as neighborhoods and defines weight functions as various aggregators on the neighborhood. GAT [20] uses the attention mechanism to learn the weight of neighbor nodes.

From the perspective of spectral domain, SpectralCNN [21] extends the convolution neural network to graph representation learning, which defines the convolution kernel in the spectral domain by using the graph signal processing theory. The convolution kernel is regarded as a trainable diagonal matrix. ChebNet [3] approximates the convolution kernel with the polynomial of the Laplacian matrix. GraphHeat [22] is a more powerful low-pass filter is designed through the heat kernel. GWNN [23] uses a wavelet basis instead of a Fourier basis to further improve the efficiency of the model.

2.2 Feature correlation graph

AM-GCN [24] uses the KNN algorithm to construct a feature graph, which can learn the most relevant information adaptively from the topology graph and node feature. SimP-GCN [25] constructs feature graph to balance structure and feature information adaptively, which can capture node similarity of original feature space and can perform well in disassortative networks.

2.3 Low-pass and high-pass signal

SpGAT[26] proposes a novel attention mechanism in spectral domain, which can effectively learn representation of low-pass and high-pass signals regarding weighted filters and graph wavelets bases.The method can make the global information of the graph be captured in an efficient manner with fewer learned parameters than GAT. From a theoretical point of view, FAGCN[18] analyzes the role of low-pass signal and high-pass signal in learning node representation, which considers that the low-pass signals can retain the similarity information of node feature, and the high-pass signals can capture the difference information of node features, the two different signals both have an important influence on the final node representation.

3 Preliminaries

We mainly study the simple graph $G = \{V, E, \mathbf{A}\}$. In a simple graph G, V represents node-set in the graph, E is the set of edges in the graph, and $\mathbf{A}$ is the adjacency matrix of the graph, where $\mathbf{A}_{ij} = \mathbf{A}_{ji}$. Let Y be the label set of all possible class and let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be the d dimensional feature matrix of all nodes in the graph. Our goal is to use the labeled node set Y^l to predict the labels of the remaining unlabeled node set Y^u.

3.1 Graph wavelet neural network

Graph wavelet neural network [23] uses wavelet basis to replace Fourier basis. Since our work is based on GWNN [23], so the following contents introduce graph wavelet neural network model.

Graph wavelet neural network defines a group of wavelet basis $\phi_s = (\phi_{s_1}, \phi_{s_2}, ..., \phi_{s_n})$, where scale is a scale parameter. A wavelet base can be written as the formula $\phi_{s_i} = U\mathbf{G}_{s_i}U^T$, $\mathbf{G}_s$ $=diag(g(s\lambda_1), g(s\lambda_2), ..., g(s\lambda_n))$ is a scaling matrix, where $g(s\lambda_i) = e^{\lambda_i s}$. The graph wavelet transform is defined as $x^* = \phi_s^{-1}x$, and the inverse wavelet transform is defined as $x = \phi_s x^*$[23, 27].

3.2 Necessity of introducing high-pass signals on the graph

In the field of computer vision, the information of the image can be expressed by different signals. The high-pass signals mainly describe the rapidly changing details in an image, such as the outline of the image [28]. Also, for non-Euclidean graph data, according to the graph signal processing theory [29], we ascend Laplacian eigenvalue, smooth change signals carried by the vector corresponding to the smaller eigenvalue, which are regarded as the low-pass signals. Sharp change signals carried by the vector corresponding to the larger eigenvalue, which are regarded as the high-pass signals. Many studies have shown that graph neural network success thanks to its ability to exploitation and utilization of low-pass signals. However, some research shows that low-pass signals only perform well in the assortative network. The effect is far inferior to the high-pass signals in the disassortative network. So low-pass or high-pass signals are both helpful to learn node representation.

4 GWNN-HF

In this section, we introduce GWNN-HF in details. As is shown in Figure 1, the purpose of this model is to obtain information of nodes with similar feature and remote in the original graph and obtain the difference between nodes with different labels while obtaining the commonality of nodes with the same label. To solve this problem, a feature correlation graph is constructed based on node feature by using the Relaxed Minimum Spanning Tree (RMST) algorithm. Based on low-pass filtering, some high-pass signals are introduced into the feature correlation graph and the original graph, respectively. The filtered representation of the original graph and the feature correlation graph is fused by the attention mechanism. Finally, the final node representation is obtained by a linear transformation of a multi-layer perceptron.

4.1 Construct a new feature correlation graph

Although graph wavelet neural networks have been proven effective in many applications, their performance can be significantly affected when the graph structure is not optimal. For example, their performance in the disassortative network is obviously reduced.

Making full use of the role of feature correlation can solve the above problems. Therefore, considering the feature similarity to build feature correlation graph can make nodes with high feature similarity are connected. In this graph, the neighbor nodes may be located far away from the target node in the original topology, may also be close to the target node. So building a feature correlation graph can capture the global information from distant nodes with similar feature in the original graph. Combining original graph and the feature correlation graph can take local and global information into account effectively. Next, we introduce the specific process of constructing feature correlation graph.

The construction of the feature correlation graph mainly consists of two steps. The first step is to calculate the similarity matrix of feature. The second step is to use the relaxed Minimum Spanning Tree (RMST) algorithm to build a feature correlation graph according to the similarity matrix.

Algorithm 1 Build feature correlation graph

```
Input: input:node feature X
Output: feature correlation adjacent matrix
1: Use $dist(C,B) = 1 - cos(C,B) = \frac{\|C\|_2\|B\|_2 - CB}{\|C\|_2\|B\|_2}$ to save cosine similarity distance between node feature, then use a similarity matrix S to save them;
2: Use Kruskal minimum spanning tree algorithm to bulid minimum spanning tree,then record $mlink_{ij}, d(i,j), d(i,i_k)\ and\ d(j,j_k)$;
3: Get feature correlation graph's adjacent matrix $A_{i,j}^{RMST}$ according to formula 1;
4: return $A_{i,j}^{RMST}$
```

Firstly, cosine similarity distance is used to calculate the distance between node feature. Then we define a similarity matrix S to store the feature similarity distance between all nodes in the whole graph.

Secondly, according to the similarity matrix S, we apply the relaxed Minimum Spanning Tree (RMST) algorithm [30] to construct the feature correlation graph.

$$\mathbf{A}_{i,j}^{RMST} = \begin{cases} 1 & if \quad d(i,j) < mlink_{ij} + \gamma(d(i,i_k) + d(j,j_k)) \\ 0 & \qquad\qquad elsewise \end{cases} \tag{1}$$

Where $\mathbf{A}_{i,j}^{RMST}$ represents the adjacency matrix of the feature correlation graph, d(i, j) represents the direct connection distance between node i and node j, i_k and j_k represents node i and node j nearest k-th neighbor respectively, $mlink_{ij} = max\{z_{i,k}, z_{k,h}, ..., z_{m,j}\}$ represents the path weights of node pairs (i, j) with the maximum weight in the multiple connected paths of the minimum spanning tree, where γ is a hyperparameter that controls the information weights of the path of the minimum spanning tree and the neighboring nodes of its two nodes. The procedure for constructing feature correlation graphs is shown in algorithm 1.

4.2 The high-pass signals are introduced for graph wavelet neural network

Since low-pass signals and high-pass signals are important, so we want to get them to enhance the representation ability of graph wavelet neural network. The following contents describe the specific process of introducing high-pass signals based on maintaining low-pass filtering in the original graph and feature correlation graph.

Firstly, according to the original graph, we can see from Figure 1, due to not being clear what type of network, we need to preserve the commonality of nodes with the same label and the difference of nodes with different labels. Therefore, we separately design the convolution kernel of the low-pass filter $\mathbf{I}$-$\epsilon_1 * \mathbf{\Lambda}$ and the convolution kernel of the high-pass filter $\mathbf{I}$+$\epsilon_1 * \mathbf{\Lambda}$ for the original graph, where ϵ_1 is the trainable parameter of the low-pass filter and high-pass filter, whose value is limited to [0,1]. The convolution kernel of low-pass filter can be rewritten by $g_\theta(\lambda_i) = 1 - \epsilon_1 * \lambda_i$, when $\epsilon_1 = 0$, $g_\theta(\lambda_i) = 1$, with the increase of λ_i, the value of $g_\theta(\lambda_i)$ is getting smaller, which can be seen to suppress high-pass signals; for the convolution kernel of high-pass filter, with the increase of λ_i, the value of $g_\theta(\lambda_i)$ increases, high-pass signals have been amplified. In this way, the convolution kernel of the low-pass filter can ensure that it only cares about the part with the smaller eigenvalues, and the corresponding eigenvector of small eigenvalue can carry the smooth-changing signals as low-pass signals. Similarly, the convolution kernel of the high-pass filter is guaranteed to only care about the part with larger eigenvalues, and the corresponding eigenvector of large eigenvalues can carry rapidly changing signals as high-pass signals. The low-pass signals' representation $\mathbf{Z}_{L_o}$ and high-pass signals' representation

$\mathbf{Z}_{H_o}$ can be obtained by the formula as follows:

$$\mathbf{Z}_{L_o} = \phi_{s_o}[\mathbf{I} - \epsilon_1 * \mathbf{\Lambda}]\phi_{s_o}^{-1} * \mathbf{X} \tag{2}$$

$$\mathbf{Z}_{H_o} = \phi_{s_o}[\mathbf{I} + \epsilon_1 * \mathbf{\Lambda}]\phi_{s_o}^{-1} * \mathbf{X} \tag{3}$$

$\phi_{s_o}, \phi_{s_o}^{-1}$ represents the wavelet basis and inverse wavelet basis of the original graph, representatively. Based on obtaining the low-pass and high-pass representation of the node feature, we set the trainable parameter α to adaptively fuse information of the two different signals. The specific fusion method in the original graph is shown in the following formula:

$$\mathbf{Z}_o = \alpha * \mathbf{Z}_{L_o} + (1 - \alpha) * \mathbf{Z}_{H_o} \tag{4}$$

Where $\mathbf{Z}_o$ represents the final representation of the original graph.

Secondly, the low-pass and high-pass signals of node feature are obtained in the same way as the original graph for the feature correlation graph. However, it should be noted that we consider the path of the minimum spanning tree and the adjacency nodes of a single node in the process of by using the Relaxed Minimum Spanning Tree algorithm, so most of the adjacent nodes in the feature correlation graph are similar, introducing too much high-pass signals may damage node representation with similar feature in the feature correlation graph, so when we fuse low-pass signals and high-pass signals of node feature, a hyperparameter β and a specially trainable parameter γ are designed to fuse the two different signals, the β is mainly used to suppress high-pass signals, the fusion method is as shown in the following formula, where $\mathbf{Z}_f$ represents the final representation obtained on the feature correlation graph.

$$\mathbf{Z}_f = \gamma * \mathbf{Z}_{L_f} + (1 - \gamma) * \beta * \mathbf{Z}_{H_f} \tag{5}$$

4.3 Fuse the representation of the original graph and feature correlation graph

At this point, we already have original feature graph representation $\mathbf{Z}_o$ and feature correlation graph representation $\mathbf{Z}_f$, then we use an attention mechanism to get the attention weight $\mathbf{K}_o, \mathbf{K}_f$ corresponding to $\mathbf{Z}_o, \mathbf{Z}_f$ to make feature correlation graph better complement the original graph.

$$(\mathbf{K}_o, \mathbf{K}_f) = attention(\mathbf{Z}_o, \mathbf{Z}_f) \tag{6}$$

Secondly, $k_o^i = softmax(k_o^i) = \frac{exp(k_o^i)}{exp(k_o^i)+exp(k_f^i)}$ was used to normalize the attention weights k_o^i and k_f^i to get the final weight values. Finally, we apply feature transformation with a layer of MLP to obtain the final representation, the formula is as follows:

$$Z = softmax(\mathbf{W} * (\mathbf{K}_o * \mathbf{Z}_o + \mathbf{K}_f * \mathbf{Z}_f) + \mathbf{B}) \tag{7}$$

Algorithm 2 The whole architecture of GWNN-HF

```
Input: node feature X
Output: the predicted result of unlabeled nodes.
Compute the cosine similarity distance of feature between nodes;
Use Algorithm 1 to build feature correlation graph;
repeat
    /*perform feature transformation*/
    Apply MLP on X;
    /*feature propagation*/
    for $i = 1$ to 2
        Design low-pass and high-pass filters for two graphs by using Equation(2)(3);
        Combine low-pass and high-pass signals for two graphs by using Equation(4)(5);
    end for
    /*get the final representation*/
    Fuse the representation of two graphs by using Equation(6)(7);
until the cross-entropy loss is minimized according to Equation(8);
```

W represents the weight matrix of the last layer of feature transformation, **B** represents the bias matrix. Thirdly,we define the cross-entroy loss function which can minimize the discrepancy between the true label and the predicted label of nodes in the graph, the specific formula is as follows.

$$\pounds = -\sum_{i \in V_l} \sum_{c=1}^{|C|} Y_{ic} \log Z_{ic} \tag{8}$$

where Y_{ic} represents the true result that node i belongs to class c. Z_{ic} represents the predicted possibility that node i belongs to class c. In this paper, we describe the architecture of GWNN-HF in pseudocode which is given in Algorithm 2.

5 Experiments

In this part, we need to evaluate the effectiveness of the GWNN-HF model in the assortative and disassortative networks. In addition, we need to answer the question:

1) How does GWNN-HF perform in the assortative network.
2) How does GWNN-HF perform in the disassortative network.
3) Influence degree of different components on GWNN-HF.

5.1 Experimental Settings

5.1.1 Datasets

Since the performance of graph neural network is different among disassortative and assortative networks, we select several representative datasets from the two different types of networks to conduct experiments. Specifically, for assortative networks, we select three popular citation networks in graph neural networks (including Cora, Citeseer and Pubmed [31]). In the citation network, the edge represents the citation relationship between two papers (undirected), the label represents the field of the paper. For disassortative networks, we choose three web datasets (including Cornell, Texas, and Wisconsin [19]), where edges represent hyperlinks between two pages, node features represent word packs in the pages. It is worth our attention that assortativity describes the measurement of connecting nodes with the same label in a dataset, and its specific formula[19] is listed as follows:

$$assortativity = \frac{1}{|V|} \sum_{v \in V} \frac{Numbers\ of\ v^{'}s\ neighbors\ who\ have\ the\ same\ label\ as\ v}{Number\ of\ v^{'}s\ neighbors} \tag{9}$$

In the assortative network, most nodes with the same label are connected, so the assortativity is relatively high. However, in the disassortative network, most of the adjacent nodes are not of the same label, so the assortativity is low. According to the value of assortativity, this paper uses Cora, CiteSeer, Pubmed as the assortative network, and Cornell, Texas, Wisconsin as the disassortative network[25]. We refer to the statistical data of these six datasets in Table 1.

Table 1 Summary of the datasets utilized in our experiments

Dataset	assortativity	Nodes	Edges	Features	Classes
Cora	0.81	2,708	5,429	1,433	7
Citeseer	0.74	3,327	4,732	3,703	6
Pubmed	0.8	19,717	44,338	500	3
Cornell	0.3	183	295	1,703	5
Texas	0.11	183	309	1,703	5
Wisconsin	0.21	251	499	1,703	5

5.1.2 Baselines

To assess the effectiveness of our model, we select the following baselines for representative semi-supervised learning.

GCN [31]: It learns node representation by aggregating information from neighbors.

GAT [20]: It is a graph neural network model that utilizes an attention mechanism to aggregate node feature.

Table 2 Node Classification accuracy(%) on assortative networks

Method	Cora	Citeseer	Pubmed
GCN [31]	81.3	71.5	79.3
GAT [20]	83.1	70.8	78.5
GWNN [23]	82.8	71.7	79.1
JK-NET [7]	80.3	68.5	78.3
GCNII [32]	82.6	68.9	78.8
SimP-GCN [25]	82.8	72.6	81.1
GWNN-HF(Ours)	**83.7**	**74.0**	**81.5**

GWNN [23]: It uses graph wavelet as a set of basis to replace the eigenvectors of Fourier basis.

JK-NET [7]: It uses dense joins to take advantage of the different neighbor ranges for each node to learn better representation.

GCNII [32]: Based on GCN, it uses residual connections and identity mapping to achieve better performance.

Gemo-GCN-S [19]: Gemo-GCN-S is a variant of Gemo-GCN which seeks to capture long-range dependencies in a disassortative network. It uses geometric relations defined in potential space to build a structural neighborhood for aggregation.

SimP-GCN [25]: It balances structure and feature information adaptively and captures the similarity of node pairs through semi-supervised learning.

It should be noted that Gemo-GCN-S is mainly designed for disassortative networks, so we only report its performance on the disassortative networks.

5.1.3 Experimental Settings

We use Pytorch to implement our approach and GPU to speed up the process of the experiment. We use Adam optimizer [33] to train the GWNN-HF with a maximum training number of 1000 epochs and a learning rate of 0.01. When patience is equals to 100, we stop the model training process in advance and select the model with the best performance according to the accuracy of the validation set. Then we run the optimal model on the test set several times to obtain the best results.

5.2 Performance Comparison

In this section, we answer the first and second question and compare the performance of our model and baselines over both disassortative and assortative networks, respectively. In order to ensure fairness, this paper directly uses the experimental results provided by other papers [25], because GWNN did not complete the experiment in the disassortative network, based on the source code provided by the author [23], we have realized the experiment of GWNN in the disassortative network.

Table 3 Node Classification accuracy (%) on disassortative networks

Method	Cornell	Texas	Wisconsin
GCN [31]	52.70	52.16	45.88
GAT [20]	54.32	58.38	49.41
GWNN [23]	81.08	81.08	86.27
JK-NET [7]	57.84	55.95	50.78
Gemo-GCN-S [19]	55.68	59.73	56.67
GCNII [32]	74.86	69.46	74.12
SimP-GCN [25]	84.05	81.62	85.49
GWNN-HF(Ours)	**89.18**	**86.48**	**94.11**

5.2.1 Comparison in the assortative network

For experiments on the assortative network, we follow the widely used semi-supervised setup, using 20 marked nodes per class for training, 500 nodes for validation, and 1000 nodes for testing. The details can be found in GCN [31]. We report the average accuracy of ten runs in Table 2, where the data in the table is the average prediction accuracy of the test set, and the highest accuracy is highlighted in each column, as for the unit of percentage. The effect of GWNN-HF shows that the original graph and feature correlation graph constructed by relaxed Minimum Spanning Tree (RMST) introduce a certain amount of high-pass signals based on low-pass signals, which can obtain better node representation. GCN, GAT and GWNN adopt different strategies to control the range of adjacent nodes and update the representation of the target node by aggregating the information of neighbor nodes in the original topology graph, without taking into account the influence of nodes that have similar features for representation of the target node. However, our model not only takes into account nodes with similar structure, but it can also take into account the nodes with similar features; both of them play a positive role in updating the representation of the target node, so that it can have a better effect on the assortative network.

5.2.2 Comparison in the disassortative network

In this section, we report the excellent performance of GWNN-HF model in three disassortative networks, such as Cornell, Texas, and Wisconsin. We follow the common design in the disassortative network, randomly divide the nodes of each class into 60%, 20% and 20% as the training set, verification set, test set and report the average accuracy of all models on more than ten randomly divided test sets.

As is shown in Table 3,which is found that GCN, GAT and GWNN can be considered as low-pass filters. The low-pass signals filtered by the low-pass filter can make the representation distance of the node smaller. However, there are many adjacent nodes with different labels near the target node in the disassortative network, which affect the node representation and lead to a poor node classification effect in the disassortative network. However, based on maintaining low-pass filtering, our model introduces additional high-pass signals, in which high-pass signals can make the representation distance of different labels nodes larger which are more suitable for disassortative networks. Therefore, the part of the low-pass signals retained by our model can make nodes with the same label more similar. In contrast, the additional high-pass signals can make the difference between nodes with different labels bigger. The overall consideration of the two components is beneficial to node representation, so our model has the best outcome in the disassortative network.

5.3 Ablation experiment

In order to better understand the effect of different modules in the model, we conduct an ablation experiment to answer the third question mentioned above. Specifically, we construct the following ablation experiments:

GWNN: Only use traditional graph wavelet neural network to the original graph.

GWNN-F: Based on the original graph, the feature correlation graph is constructed according to the node feature, we apply graph wavelet neural network with low-pass filtering property on the original graph and feature correlation graph.

GWNN-HF: Based on the original graph, the feature correlation graph is constructed based on node feature, we apply graph wavelet neural network and add additional high pass signals to the original graph and feature correlation graph, which is also the model we designed above.

Figure 2 and Figure 3 show the experimental results of each component in assortative and disassortative networks. According to the experimental results, the method proposed by us, under the consideration of all components, which achieves the best results in both disassortative and assortative networks.

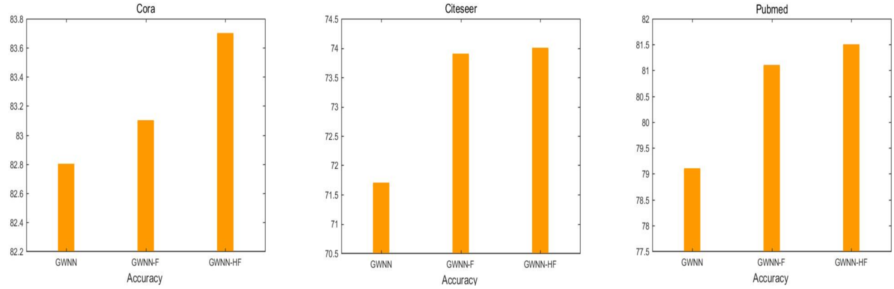

Fig. 2 Comparison of assortative network.

It is worthwhile to note that adding feature correlation graph in an assortative network can not only obtain the neighbor node information of the original topology graph, but also obtain the node information of remote and feature similar in the original topology graph, to a certain extent, which can enhance aggregation ability of similar nodes for graph wavelet neural network in the assortative network. From the analysis of the CiteSeer dataset in Figure 2, we can find that the performance of GWNN-F has been greatly improved compared with GWNN, because it shows that utilizes feature correlation can improve node representation in Citeseer dataset where there may be a large number of the distant but similar node.

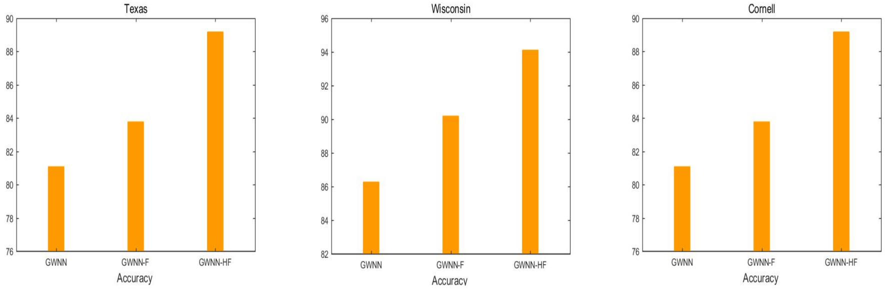

Fig. 3 Comparison of disassortative network.

In the disassortative network, it can be found that adding feature correlation graph component can improve representation learning ability of graph wavelet neural network. However, We easily find from the dataset of three disassortative networks in Figure 3, after introducing high-pass signals to the graph wavelet neural network, it can make the nodes with different labels easier to distinguish in the training process, and its algorithm performance improves more.

6 Conclusion

Throughout this paper, we put forward the GWNN-HF model for semi-supervised node classification. Firstly, we construct additional feature correlation graph and use the attention mechanism to fuse the representation of the original graph and feature correlation graph. Secondly, we introduce high-pass signals, not only considering the similarity of the node with the same label, but also considering difference of nodes with different labels. Several experiments show that GWNN-HF is superior to existing graph neural network methods.

7 Acknowledge

This work is supported by the National Natural Science Foundation of China (61876001), the Natural Science Foundation for the Higher Education Institutions of Anhui Province of China (KJ2021A0039).

References

[1] Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model cnns. In: CVPR, pp. 5425–5434 (2017)

[2] Guha, S., Wicker, S.B.: Do birds of a feather watch each other?: Homophily and social surveillance in location based social networks. In: CSCW, pp. 1010–1020 (2015)

[3] Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: NeurIPS, pp. 3837–3845 (2016)

[4] Bojchevski, A., Klicpera, J., Perozzi, B., Kapoor, A., Blais, M., Rózemberczki, B., Lukasik, M., Günnemann, S.: Scaling graph neural networks with approximate pagerank. In: KDD, pp. 2464–2473 (2020)

[5] Jia, J., Benson, A.R.: Outcome correlation in graph neural network regression. CoRR **abs/2002.08274** (2020)

[6] Klicpera, J., Bojchevski, A., Günnemann, S.: Predict then propagate: Graph neural networks meet personalized pagerank. In: ICLR (2019)

[7] Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K., Jegelka, S.: Representation learning on graphs with jumping knowledge networks. In: ICML, pp. 5449–5458 (2018)

[8] Hamilton, W.L., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NeurIPS, pp. 1024–1034 (2017)

[9] Gao, H., Wang, Z., Ji, S.: Large-scale learnable graph convolutional networks. In: KDD, pp. 1416–1424 (2018)

[10] Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: ICLR (2019)

[11] Abu-El-Haija, S., Perozzi, B., Al-Rfou, R., Alemi, A.A.: Watch your step: Learning node embeddings via graph attention. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) NeurIPS, pp. 9198–9208 (2018)

[12] Wang, J., Deng, Z.: A deep graph wavelet convolutional neural network for semi-supervised node classification. In: IJCNN, pp. 1–8 (2021)

[13] Zhang, M., Dong, B., Li, Q.: Ms-gwnn: Multi-scale graph wavelet neural network for breast cancer diagnosis. In: ISBI, pp. 1–5 (2022)

[14] Huang, W., Zhang, T., Rong, Y., Huang, J.: Adaptive sampling towards fast graph representation learning. In: NeurIPS, pp. 4563–4572 (2018)

[15] Chen, Y., Wu, L., Zaki, M.J.: Iterative deep graph learning for graph neural networks: Better and robust node embeddings. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) NeurIPS (2020)

[16] Abu-El-Haija, S., Perozzi, B., Kapoor, A., Alipourfard, N., Lerman, K., Harutyunyan, H., Steeg, G.V., Galstyan, A.: Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In: ICML, pp. 21–29 (2019)

[17] Wu, F., Jr., A.H.S., Zhang, T., Fifty, C., Yu, T., Weinberger, K.Q.: Simplifying graph convolutional networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) ICML, pp. 6861–6871 (2019)

[18] Bo, D., Wang, X., Shi, C., Shen, H.: Beyond low-frequency information in graph convolutional networks. In: AAAI, pp. 3950–3957 (2021)

[19] Pei, H., Wei, B., Chang, K.C., Lei, Y., Yang, B.: Geom-gcn: Geometric graph convolutional networks. In: ICLR (2020)

[20] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: ICLR (2018)

[21] Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. In: NeurIPS, pp. 15509–15519 (2019)

[22] Xu, B., Shen, H., Cao, Q., Cen, K., Cheng, X.: Graph convolutional networks using heat kernel for semi-supervised learning. In: IJCAI, pp. 1928–1934 (2019)

[23] Xu, B., Shen, H., Cao, Q., Qiu, Y., Cheng, X.: Graph wavelet neural network. In: ICLR (2019)

[24] Wang, X., Zhu, M., Bo, D., Cui, P., Shi, C., Pei, J.: AM-GCN: adaptive multi-channel graph convolutional networks. In: KDD, pp. 1243–1253 (2020)

[25] Jin, W., Derr, T., Wang, Y., Ma, Y., Liu, Z., Tang, J.: Node similarity preserving graph convolutional networks. In: WSDM, pp. 148–156 (2021)

[26] Chang, H., Rong, Y., Xu, T., Huang, W., Sojoudi, S., Huang, J., Zhu, W.: Spectral graph attention network with fast eigen-approximation. In: CIKM, pp. 2905–2909 (2021)

[27] Zheng, W., Qian, F., Zhao, S., Zhang, Y.: M-GWNN: multi-granularity

graph wavelet neural networks for semi-supervised node classification. Neurocomputing, 524–537 (2021)

[28] Chen, Y., Fan, H., Xu, B., Yan, Z., Kalantidis, Y., Rohrbach, M., Yan, S., Feng, J.: Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In: ICCV, pp. 3434–3443 (2019)

[29] Ortega, A., Frossard, P., Kovacevic, J., Moura, J.M.F., Vandergheynst, P.: Graph signal processing: Overview, challenges, and applications. Proc. IEEE, 808–828 (2018)

[30] Beguerisse-Díaz, M., Vangelov, B., Barahona, M.: Finding role communities in directed networks using role-based similarity, markov stability and the relaxed minimum spanning tree. In: GlobalSIP, pp. 937–940 (2013)

[31] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)

[32] Chen, M., Wei, Z., Huang, Z., Ding, B., Li, Y.: Simple and deep graph convolutional networks. In: ICML, pp. 1725–1735 (2020)

[33] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)