GSpect: Spectral Filtering for Cross-Scale Graph Classification

Xiaoyu Zhang, Wenchuan Yang, Jiawei Feng, Bitao Dai, Tianci Bu, and Xin Lu

***Abstract*—Identifying structures in common forms the basis for networked systems design and optimization. However, real structures represented by graphs are often of varying sizes, leading to the low accuracy of traditional graph classification methods. These graphs are called cross-scale graphs. To overcome this limitation, in this study, we propose GSpect, an advanced spectral graph filtering model for cross-scale graph classification tasks. Compared with other methods, we use graph wavelet neural networks for the convolution layer of the model, which aggregates multi-scale messages to generate graph representations. We design a spectral-pooling layer which aggregates nodes to one node to reduce the cross-scale graphs to the same size. We collect and construct the cross-scale benchmark data set, MSG (Multi Scale Graphs). Experiments reveal that, on open data sets, GSpect improves the performance of classification accuracy by 1.62% on average, and for a maximum of 3.33% on PROTEINS. On MSG, GSpect improves the performance of classification accuracy by 13.38% on average. GSpect fills the gap in cross-scale graph classification studies and has potential to provide assistance in application research like diagnosis of brain disease by predicting the brain network's label and developing new drugs with molecular structures learned from their counterparts in other systems.**

***Index Terms*—complex networks, graph neural networks, graph classification, cross-scale, spectral graph theory**

I. INTRODUCTION

DATA that have a non-Euclid structure—such as protein structures [1], social networks [2] and compounds [3]—are often represented by graphs with nodes and edges. As structure determines function in many networked systems, graph classification (for exact definition, please refer to III-A) is a fundamental research problem in numerous fields. For example, in computer vision, graph classification methods are used to measure the similarity of human action recognition among graphs [4]. In neuroscience, researchers use graph classification methods to study the similarity of brain networks [5]. In chemistry, graph classification methods are used to learn the similarity of chemical compounds in terms of their effect on reaction partners [6]. Fields such as bioinformatics and molecular chemistry often encounter a problem named graph classification: graphs with different structures possess totally different functions. Researchers must separate graphs with different structures to select appropriate graphs in a short time. For example, Alzheimer's disease (AD) is known to be caused by structural changes in the brain. Researchers take samples of brain networks and determine whether the sample is likely to develop AD [7].

Researchers have proposed numerous methods to accomplish the graph classification problem, like graph kernels [8]. However, these methods often define graphs in a heuristic manner, thereby resulting in low explainability and flexibility of these methods. It is for this reason that graph neural networks (GNNs) have become a popular method for graph classification tasks in recent years due to their ability to learn node and edge representations and capture messages from complex graph structures. Researchers usually design a GNN convolution layer to obtain the graph representation and design a GNN-based pooling layer to reduce the size of the graphs. One of the most classic definition of GNN convolution layers is spectral-based GNN. Spectral-based GNNs utilize diagonal spectral filters to capture information from the graph's spectrum. This method has been wildly used for node classification and edge prediction tasks [9] [10]. For instance, [11], [12] focus on improving traditional spectral filtering methods by introducing node-oriented and diverse spectral filtering approaches which address limitations such as the constrained expressive power of conventional spectral filtering and the disregard for regional heterogeneity common in real-world networks, significantly enhancing baseline performances. [13] primarily explores the theory of vertex-frequency analysis on graphs. However, spectral-based methods have a few limitations [14]: First, any perturbation to the graph results in the change of the graph's eigenvalues. Second, the learned filters are size-dependent. One graph determines a unique network structure, which implies that it is difficult to be applied to graphs with different sizes. So it is difficult to be used in the cross-scale graph classification tasks.

Traditional graph classification methods only work on comparing structure of similar sizes [15] [16] [17]. However, in practice, structures of an order-of-magnitude difference in size may have the same function. For example, in biology, the structure of proteins, which possesses critical functions—such as immune signaling [18], targeted therapeutics [19], sense-response systems [20] and self-assembly materials [21]—can be represented as graphs whose nodes represent the atoms and edges represent the chemical bonds. The protein structure determines its function. Certain proteins which have the same functions usually have similar graph structures. However,

This work was supported by the National Natural Science Foundation of China (72025405, 72088101), the National Social Science Foundation of China (22ZDA102), the Hunan Science and Technology Plan Project (2020TP1013, 2020JJ4673, 2023JJ40685), the Shenzhen Basic Research Project for Development of Science and Technology (20200829172650001), and the Innovation Team Project of Colleges in Guangdong Province (2020KCXTD040). The authors declare that they have no conflict of interest. (Corresponding author: Xin Lu.)

Email addresses: xiaoyuzhang_2023@163.com (Xiaoyu Zhang), wenchuanyang97@163.com (Wenchuan Yang), fengjiawei126@gmail.com (Jiawei Feng), daibitao@nudt.edu.cn (Bitao Dai), btc010001@gmail.com (Tianci Bu), xin.lu.lab@outlook.com (Xin Lu)

these protein-graphs occasionally have an order-of-magnitude difference in the number of nodes [22]. This set of graphs is called cross-scale graphs. Cross-scale graph classification tasks refers to dividing the graphs with an order-of-magnitude difference in the number of nodes into sets. Research on cross-scale graphs is an important research direction in the field of complex networks. Cross-scale graphs plays an important role in the practical applications such as network clustering [23], hierarchical reduction [24], and state partition [25]. Researchers require cross-scale graph classification algorithms to select structure-similar but cross-scale proteins from a huge selection space. [26] have proposed methods tailored to datasets of varying graph sizes, but these studies have exclusively designed methods for small-scale, sparse graphs (don't consider large-scale graphs) and conducted experiments only on publicly available datasets with similar graph sizes. There is no available method for cross-scale graphs' classification task and there are no open data sets with graphs which have enough difference (up to 10^3) in the number of nodes.

Graph Wavelet Transform (GWT) is a powerful tool for capturing multi-scale graph representations [27] [28] due to its unique properties, making it becomes a powerful tool to solve cross-scale graph classification problems. The advantages of GWT is listed as follows. GWT offers multi-scale analysis capabilities, effectively representing both local and global features of graph structures [27]. Besides, its localization properties in both spatial and frequency domains enable efficient capture of local structural information [29]. In addition, GWT typically produces sparse representations of graph signals, facilitating key feature extraction [30]. Compared to global spectral methods, GWT often demonstrates higher computational efficiency, especially for large-scale graphs [31]. Apart from that, it naturally adapts to irregular graph structures, a challenge for traditional wavelet transforms [32]. Recently, it is proved that GWT allows for cross-scale information integration, helping to capture hierarchical structures in graphs [33]. Moreover, it exhibits robustness to minor structural changes, which is valuable when dealing with noisy data [28]. These characteristics make GWT a versatile and effective tool for multi-scale graph representation, with wide applications in graph classification, node classification, and graph signal processing.

In this article, we modified the spectral-based GNN using the graph wavelet theory and design a novel framework (GSpect) to accomplish cross-scale graph classification tasks. Specifically, considering the characteristic that the wavelet function can accurately capture the signal information in different frequency bands, we first use a graph wavelet neural network as the convolution layer for graph classification tasks. Second, we design a graph pooling layer. Compared with other spectral clustering methods [34] [29], we directly perform Fourier transformation on the graph's adjacency matrix and node attributes directly to obtain the frequency domain representation. We use spectral filters to filter high-frequency information and resize the graph on the principle of the save-most message. Third, considering the fact that there are no appropriate cross-scale graph classification data sets, we collect three classes of empirical networks—covering the set of protein structure data, macromolecular compound structure data, and social networks in combination with the three typical modeled networks of ER [35], WS [36] and BA [37], to create a synthesis cross-scale graph classification benchmark data set MSG. We verify the performance of GSpect both on the open data sets and on MSG.

This article makes the following contributions:

1. We apply the graph wavelet theory to graph classification tasks and use the graph wavelet convolution layer to aggregate multi-scale information from graphs and generate graph-level representation.

2. We design a pooling layer by using non-square learnable filters in the frequency domain to filter unusable messages and generate a low-order graph (refers to a graph structure obtained by aggregating or filtering nodes, resulting in a structure with fewer nodes and edges).

3. We collect cross-scale graph data and generate a cross-scale graph data set MSG and conduct experiments on both open data sets and MSG. We test the classification accuracy and the results indicate that on open data sets, GSpect improves the performance of classification accuracy by 1.62% on average, and for a maximum of 3.33% on PROTEINS. On MSG, GSpect improves the performance of classification accuracy by 13.38% on average of all state-of-art comparative models.

The remainder of this article is organized in the following manner: Section II presents the related works. Section III proposes GSpect in detail. Section IV presents the experimental results, including the comparison experiment, ablation study, and sensitivity analysis. Section V summarizes our contributions and future directions.

II. RELATED WORKS

A. Graph Kernel Models for Graph Classification

Graph kernels capture the similarity between graphs for graph classification tasks. Given a set of graphs, the graph kernel methods aim to learn the kernel function that captures the similarity between any two graphs. Traditional graph kernels, such as random walk kernel, subtree kernel, and shortest-path kernels are widely used in graph classification tasks [8] [38]. The WL algorithm [39] maps the original graph to a sequence of graphs whose node attributes are generated from graph topology and label information. A kernel family, including an efficient kernel family of comparison subtree patterns, can be defined from this WL sequence. This algorithm has became one of the most widely used graph kernel methods for graph classification. Al-Rfou et al. [40] proposed deep divergence graph kernels (DDGK). DDGK learn kernel functions for a pair of graphs. Given two graphs G_1 and G_2, this method learns a kernel function $K(.)$ as a similarity metric function for graphs. The function is defined in the following manner:

$$k\left(G_1, G_2\right) = \left\|\Psi\left(G_1\right) - \Psi\left(G_2\right)\right\|^2, \tag{1}$$

where $\Psi\left(G_1\right)$ is the graph representation of G_1. This method learn the graph representation by computing the divergence of the target graph. Given a set of source graphs $G_1, G_2, ..., G_N$, a graph encoder is the representation of each graph in the set.

Then, for the target graph G_i, the divergence between G_i and the source graph is computed to measure the similarity. The equation of divergence between G_a and G_b is as given bellow:

$$\mathcal{D}'\left(G_a \| G_b\right) = \sum_{v_i \in V_a} \sum_{j, e_{ij} \in E_a} -\log \Pr\left(v_j \mid v_i, H_b\right), \quad (2)$$

where a is the encoder trained on graph G_a. $D'(G_a \parallel G_b$ represents the divergence from graph G_a to graph G_b. $\Pr(v_j|v_i, H_b)$ represents the probability of node v_j occurring given node v_i under the encoder H_b of graph G_b.

However, graph kernel models have a few limitations: Most of them have low computational efficiency, and graph kernel methods use kernel functions(like Equation 1) to measure the similarity between two graphs, which implies that graph kernel methods can't be used to handle graph classification problems with a lot of graphs.

B. Classic GNN Models for Graph Classification

In recent years, researchers have become increasingly interested in the extension of the deep learning method to graphs. Driven by the success of deep neural networks, the researchers drew on the ideas of convolutional neural networks, recurrent neural networks, and auto encoder to define and design a neural network structure for processing graph data. Consequently, a new method called GNNs emerged. Researchers have designed a few GNN-based graph classification methods. For example, the graph convolutional network (GCN) [9] is one of the earliest methods in this discipline. GCNs learn node representations and propagate them to other nodes using a spectral graph convolution technique. In many graph classification tasks, GCN have demonstrated state-of-the-art performance. However, GCNs have limitations in capturing long-range relationships and higher-order graph structures. To solve these problems, MPNN [41] applies a message-passing algorithm to learn node representations on the local graph structure. It has been demonstrated that MPNN are efficient in capturing higher-order graph topology and long-range relationships. With the development of the attention mechanism, graph attention networks (GATs) [42] have become a popular method for graph classification. GATs use self-attention to learn node representations, thereby enabling the model to focus on only the key nodes in the graph. GATs have been shown to achieve state-of-the-art performance on many graph classification tasks. In addition to these methods, several other GNN variants have been proposed, such as graph isomorphism networks (GINs) [43]. These techniques have improved the classification accuracy for a variety of graph classification problems.

As the size of graphs to be classified are usually different and cannot be directly compared, many methods apply graph pooling to resize the graphs to a unique size before the classification. A number of intuitive methods are used for graph pooling. For example, max-pooling and mean-pooling use the maximum or average value of a group of nodes to represent them [44]. However, these methods lack flexibility, which reduces the competitiveness of these methods. To overcome these limitations, Ma et al. [45] introduced EigenPooling, an innovative approach rooted in the graph Fourier transform. This method leverages the spectral domain to effectively pool nodes in a graph. However, the pooling process is heuristic and cannot be optimised by machine learning algorithms. For this problem, currently available spectral clustering (SC) methods [34] [46] were proposed to identify clusters, which are subsets of nodes that are more densely connected to each other than to the rest of the graph. However, it leads to more computational complexity. However, these methods only execute pooling once, which occasionally leads to the loss of key nodes. To solve this problem, Ying et al. [47] developed the hierarchical pooling approach (DiffPool). They created the concept assign matrix that maps a set of nodes to a single node using GNN models. The function of the assignment matrix is given below:

$$S^{(k)} = softmax[GNN_{k,pooling}(A^{(k)}, X^{(k)})], \quad (3)$$

where $A^{(k)}$ and $X^{(k)}$ are the graph's adjacency matrix and graph representation matrix. $GNN_{k,pooling}$ is a learnable function. In practice, DiffPool combines its pooling method with the differentiable graph encoder to make the architecture top-to-end trainable.

C. Wavelet Transform-Based Research

As a part of the spectral theory, the wavelet theory has been widely used in the field of image processing and signal analysis. For example, Yahia et al. [48] use wavelet neural networks for image classification and attain high accuracy.

Some researchers applied wavelet transform to the spectral graph theory. For example, Hammond et al. [27] defined a wavelet function to project the graph to the wavelet domain, the equation is defined in the following manner:

$$\psi_{s,i}(j) = \sum_{t=1}^{N} g\left(f\lambda_t\right) u_t^{\star}(i) u_t(j), \quad (4)$$

where N is the number of vertices, λ_t is the t-th eigenvalue of the graph Laplacian matrix, u_t is the eigenvector of the Laplacian matrix. The symbol $\star$ denotes the complex conjugate operator, and g is the spectral graph wavelet generating kernel. This research is the first to propose the concept of the graph wavelet transform. However, it does not combine graph wavelet theory and deep learning.

Graph wavelet transform is becoming more frequently used in the design of GNNs. Xu et al. [31] designed the graph wavelet neural network(GWNN) using spectral graph theory for node classification tasks and obtained satisfactory results. They reported that using graph wavelet transform can circumvent the short-comings of previous spectral CNN methods, depending on the graph Fourier transform. Similarly to the graph Fourier transform, the wavelet base is designed in the following manner:

$$\Psi_s = \mathbf{U_s} \mathbf{G_s} \mathbf{U_s^T}, \quad (5)$$

where $\mathbf{U_s}$ represents the Laplacian eigenvectors and $\mathbf{G_s} = diag(e^{\lambda_1 s}, ..., e^{\lambda_n s})$ is the scaling matrix. Substituting the graph Fourier transform with wavelet transform, GWNN uses

diagonal masks to generate the representation of each node. The structure of the m-th layer is defined as:

$$\boldsymbol{X}_{[:,j]}^{m+1} = h\left(\psi_s \sum_{i=1}^{p} \boldsymbol{F}_{i,j}^{m} \psi_s^{-1} \boldsymbol{X}_{[:,i]}^{m}\right) \quad j = 1, \cdots, q. \tag{6}$$

Note that $F_{i,j}^{m}$ is a diagonal matrix, which is effective for node-level classification tasks, as the features of different nodes cannot be mixed. GWNN is highly competitive at node-level tasks. However, GWNNs don't have a mechanism to handle graphs of different sizes, which is crucial for graph classification tasks. Besides, GWNNs lack a standard pooling mechanism to aggregate node-level features into a fixed-size graph-level representation, which is necessary for classifying graphs of varying sizes.

As the application in graph multi-modal learning, Behmanesh et al. [49] proposed a graph wavelet convolution network (GWCN) for multi-modal learning. GWCN generates single-modal representations by applying the multi-scale graph wavelet transform and learning permutations that encode correlations among various modalities. GWCN have the best performance on node classification tasks.

Wavelet-based methods are a powerful tool for capturing multi-scale graph representations. However, currently, few methods use graph wavelet transform for cross-scale graph classification.

III. GSpect

A. Problem Description

Let $G = \{V, E\}$ represent a graph, with V and E being the set of nodes and edges, respectively. $A \in \{0,1\}^{n \times n}$ represents the adjacency matrix and $X \in \mathbb{R}^{n \times l}$ represent the node attribute matrix. l represents the length of the attribute vector. There is a set of labeled graphs $(\{G\}, \{y\})$, where $y_i \in \mathbb{Z}$ represents the label of G_i, and max[size($\{G\}$)]/min[size($\{G\}$)] $\geq 10^3$. The target of the cross-scale graph classification task is to learn a mapping $G \rightarrow y$. Compared with other machine learning methods applied in computer vision and natural language processing, we need to convert graphs with different topologies into vector $v \in \mathbb{R}^q$, where $q \leq min(n)$. Then, the mature approach of machine learning methods can be used. Fig. 1 depicts an example of cross-scale graph classification.

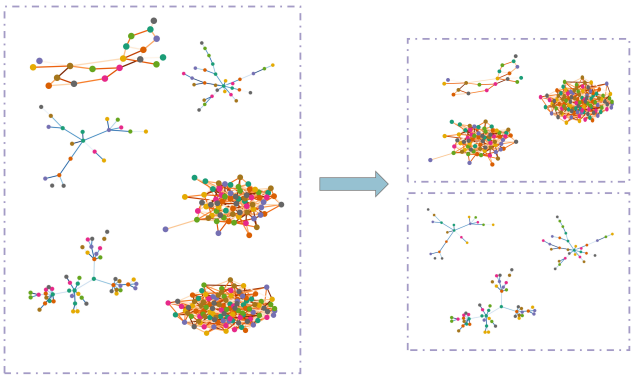


Fig. 1. An example of cross-scale graph classification

B. Model Framework

In this section, we introduce the framework of our model GSpect. GSpect consists of four parts(Fig. 2). The first part is the convolution layer. We use graph wavelet transform for the convolution layer to generate the graph-level representation. In the second part, we design the spectral-pooling layer to filter the useless information and obtain the low-order representation for classification. The spectral-pooling layer aggregates the nodes with similar representations in the spectrum and obtain a low-order graph. The third part is a fully connected layer for classification. Because the convolution and pooling process need to be repeated many times, we use simple GCN in convolution after pooling. Finally we design an optimising function to optimise the model. Furthermore, we proved the stability of the model (see Appendix A). In addition, for the effectiveness and information loss of the model, please refer to Appendix C and D, respectively.

C. Data Renomorlization

Due to the differences in graph sizes, it is challenging to integrate them into a unified computational framework. Therefore, similar to many previous methods, this paper first performs a renormalization of the data.

For all graphs' adjacency matrices $\{A_1, A_2, \ldots, A_t, \ldots\}$, where $A_t \in \mathbb{R}^{n_t \times n_t}$, let $n' = max\{n_1, n_2, \ldots, n_t, \ldots\}$, then for each matrix A_t, the renomorlized matrix $A_t^{'}$ is defined as follows:

$$A_t'(i,j) = \begin{cases} A_t(i,j) & \text{if } i < n^t \text{ and } j < n^t \\ 0 & \text{otherwise} \end{cases}, \tag{7}$$

where $n' = max\{n_1, n_2, \ldots, n_t, \ldots\}$.

Similarly, for attribute matrices for graphs $\{X_1, X_2, \ldots, X_t, \ldots\}$, where each feature matrix $x_t \in \mathbb{R}^{n_t \times l}$. Let $n' = \max\{n_1, n_2, \ldots, n_t, \ldots\}$. Then, for each feature matrix X_t, the renormalized matrix X_t' is defined as:

$$X_t'(i,j) = \begin{cases} X_t(i,j), & \text{if } k < n_t \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $n' = \max\{n_1, n_2, \ldots, n_t, \ldots\}$.

After processing, all the data have the same dimensions, facilitating subsequent operations.

D. Graph Wavelet Convolution Layer

As the first step of GSpect, we design a convolution layer to generate the graph presentations. For traditional convolution methods, cross-scale graphs has big difference in size, which leads to the difficulty of getting graph presentations. In this section, we propose the graph wavelet convolution layer (GWC). Taking advantage of the fact that the wavelet function can capture multi-scale messages, we use the wavelet transform to project the graph into the wavelet domain and use a learnable filter to aggregate messages from every entry and obtain the graph representation.

In earlier research, wavelet bases are defined in the following manner: $\Psi_f = [\psi_{f,1}, \ldots, \psi_{f,N}]$, where $\psi_{f,i}$ represent the

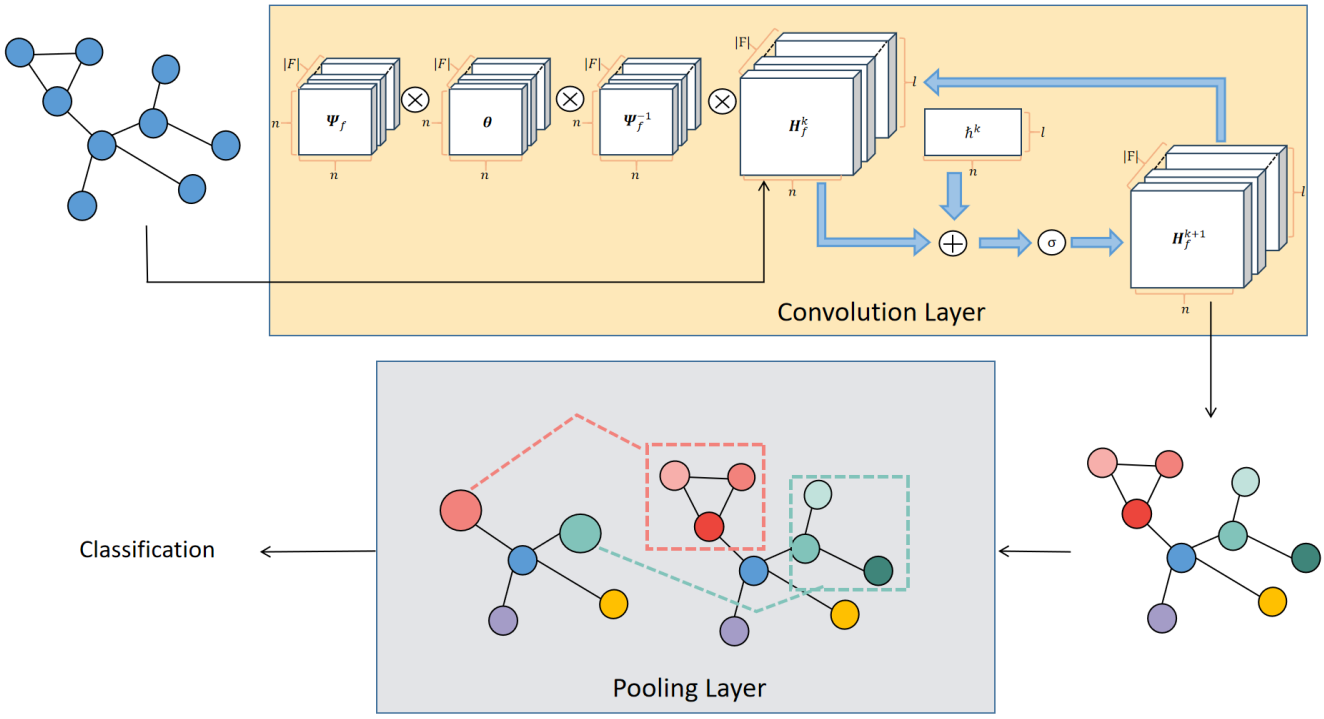


Fig. 2. The model structure of GSpect. GSpect consists of four phases. The first phase consists of the convolution layers. Each layer has F multi-scale graph wavelet convolution. The second phase is a pooling layer. This layer aggregates the nodes with similar representations in the spectrum and yields a low-order graph. The third phase is a full-connect layer for classification. The different colors indicate the feature vectors of nodes. In the second phase, nodes with similar features (depicted as the same color in the diagram) are aggregated into a single node.

the transform matrix at node i and scale f. Different studies have various definitions of $\psi_{f,i}$. A few traditional functions of wavelet bases need to compute the eigenvalues of the graph, which leads to a large amount of computation. To escape this, we use the definition of [27] to approximate the wavelet bases, which is defined in the following manner:

$$\Psi_f = \frac{1}{2}c_{0,f} + \sum_{i=1}^{M} c_{i,f}\mathbf{T}_i(\tilde{\mathbf{L}}), \tag{9}$$

$$c_{i,f} = 2e^{-f}J_i(-f), \tag{10}$$

where $\tilde{\mathbf{L}}$ is the Chebyshev polynomial [17] of order i which is used to approximate Ψ_f, M is the number of Chebyshev polynomials and $J_i(-f)$ is the Bessel function of the first category [50] and f is the wavelet scale.

According to prior research [49], we use the wavelet base Ψ_f^{-1} to project the graph's embedding matrix to the wavelet domain.

Since the formula Equation 10 is in an approximate form, the inverse of the matrix may not exist. Therefore, this article uses the pseudoinverse of the matrix instead. We first perform singular value decomposition on Ψ_f, that is:

$$\Psi_f = V\Sigma U^T. \tag{11}$$

Where V and U are left and right singular vector matrix. Then the inverse Ψ_f^{-1} is defined as follows:

$$\Psi_f^{-1} = V\Sigma^{-1}U^T. \tag{12}$$

Then, in the wavelet domain we use a learnable filter to aggregate messages from every entry. Thereafter, we use Ψ_f to convert the representation back. Finally, we use the bias and activation functions to formalise the convolution layer. The one-scale-channel convolution layer is defined in the following manner:

$$H^{k+1}_{n'\times l,f} = \sigma(\Psi_{n'\times n',f}\Theta_{n'\times n'}\Psi^{-1}_{n'\times n',f}H^{k}_{n'\times l,f} + \hbar_{n'\times l}), \tag{13}$$

where n' represents the node number, f represents the wavelet scale, and k represents the k-th layer. Θ and $\hbar$ are learnable parameters. There are many scales which are responsible for aggregating messages on their own scale. By averaging the messages of F scales, the total convolution layer is defined in the following manner:

$$H^{k}_{n'\times l} = \frac{1}{F}\sum_{f=1}^{F} H^{k}_{n'\times l,f}. \tag{14}$$

We use average graph representation by averaging the graph representations of all scales, which synthesise the graph structure messages on different scales.

E. Spectral-pooling Layer

Since the graphs have different sizes even after convolution, they cannot be directly classified. To solve this problem, we

design a pooling layer to process the graph presentation and generate graphs in the same size for classification.

Motivated by the research [45] [47], we continue to use the concept assignment matrix:

$$S^k = GNN_{k,pooling}(A^k, X^k), \tag{15}$$

which implies learning a project matrix that projects the adjacency matrix to a low-order adjacency matrix. In essence, it converges a group of nodes to a single node. Rather than using a normal GNN structure to learn S^k directly, we propose a new method in this article. We use Fourier transform to convert the adjacency matrix A and graph embedding X into a frequency domain and use a spectral filter to filter out useless information and reduce the size of matrix through spectral convolution. The assign matrix is defined in the following manner:

$$S^k_{(n'-m)\times n'} = \xi^k_{(n'-m)\times(n'-m)}\theta^k_{(n'-m)\times n'}\xi^{-1,k}_{n'\times n'}. \tag{16}$$

where n' and m is the node number before pooling and after. $\theta_{(n'-m)\times n'}$ is the learnable parameter. The formula of $\xi_{N\times N}$ is defined as follows:

$$\xi_{N\times N} = \frac{1}{\sqrt{N}}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix}, \tag{17}$$

where $\omega = e^{-2\pi i/N}$ is N the complex $N-th$ root of unity, and i is the imaginary unit.

Thus, the total equation of adjacency matrix A and graph embedding X is:

$$X'^{k+1}_{(n'-m)\times l} = S_{(n'-m)\times n'}X'^{k}_{n'\times l}, \tag{18}$$

$$A'^{k+1}_{(n'-m)\times(n'-m)} = S^k_{(n'-m)\times n'}A'^{k}_{n'\times n'}(S^k_{(n'-m)\times n'})^T. \tag{19}$$

F. The Optimization Method

The parameters in the model need to be optimized. In this section, we introduce the optimization function of GSpect. According to existing research [51], it is difficult to optimize the model using gradient descent only during the graph classification task. To solve this question, we use the weighted optimization function. We will introduce the optimization functions separately.

Cross entropy is an important concept in information theory. Its value represents the difference between two probability distributions. The approximate of the target probability distribution can be obtained by minimizing cross entropy. First, we use the cross entropy function as a part of our optimization function, which is defined in the following manner:

$$L_\varepsilon(p,q) = \frac{1}{c}\sum_{i=1}^{c} p_i log(q_i), \tag{20}$$

where p_i and q_i are true and predicted labels, and c is the class number.

The assign matrix should meet one condition: the nodes having strong links have higher probability of aggregating to a new node [47]. Thus the second part of the optimization function is expressed in the following manner:

$$L_p = \left\|A^k - S^k(S^k)^T\right\|_F r, \tag{21}$$

where $\|.\|_F r$ represents the Frobenius norm. This equation implies to let A^k and $S^k(S^k)^T$ be as close as possible. Specifically, for A^k, when $k=0$, A^0 is the graph's adjacency matrix. When $k \neq 0$, $A^k$is the processed adjacency matrix in the k-th layer. $A^{(k)}_{ij}$ represents the link between node i and node j in the k-th layer.

For $S^{(k)}S^{(k)^T}$, $S^{(k)} \in \mathbb{R}^{n_k\times n_{k+1}}$ $(n_k > n_{k+1})$ is the probability matrix, $S^{(k)}_{ir}$ represents the probability of node i in the k-th layer, thereby mapping to node j from cluster r in $(l+1)$-th layer. When the probability of two nodes mapping to one cluster increases, the value of $S^{(l)}S^{(l)^T}$ becomes larger. Minimizing L_p and retaining the correct assignment matrix $S^{(l)}$ can let the pair of nodes that has a stronger link easily map to one cluster.

Thus, the total optimisation function is expressed in the following manner:

$$L_t = (1-\beta)L_\varepsilon + \beta L_p, \tag{22}$$

where β is the equilibrium coefficient.

IV. EXPERIMENT

In this section, we test the model's effectiveness on graph classification tasks. We aim to answer the following questions:

$Q1$ How does our model compared to other advanced models in open data sets?

$Q2$ To what extent does our model improve the performance of a baseline GNN?

$Q3$ Is GSpect sensitive to changes in hyperparameters?

The code and other materials are available at https://github.com/XiaoyuZhang001/GSpect.

A. Experiment Settings

1) Data Sets: We use the following five open data sets to verify the effectiveness of the model:

D&D [15] (Biological macromolecules). D&D is a protein data set. It extracted 1178 high-resolution proteins from a non-redundant subset of the protein database using simple features, such as secondary structure content, amino acid propensity, surface properties, and ligands. The nodes are amino acids, and if the distance between the two nodes is less than six angstroms, an edge is used to represent this relationship. Nodes in DD data set are unlabeled and nodes only have features. The criterion for classification is whether a protein is an enzyme.

PTC [52] (Small molecules). PTC is a collection of 344 compounds that report carcinogenicity to rats. Researchers need to classify these compounds to the criterion of carcinogenicity. Nodes represent atoms and edges between nodes represent bonds between corresponding atoms. Each node has 19 node labels.

PROTEINS [53] (Biological macromolecules). PROTEINS is another network of proteins. The task is to determine whether such molecules are enzymes. The nodes are amino acids.

TABLE I
BASIC STATISTICS OF THE OPEN DATA SETS AND MSG

Name	Avg Graph Size	Avg Degree	Avg Edges Number	Min Max Graph Size	S.D. of Node Distribution	Avg Shortest Path
PTC	25.56	1.99	25.25	[2, 109]	16.25	8.78
MUTAG	17.93	2.19	19.79	[10, 28]	4.58	8.22
PROTEINS	39.05	3.73	57.72	[4, 620]	45.76	10.75
D&D	268.70	4.98	173.98	[30, 903]	161.33	12.14
IMDB-B	19.77	8.89	95.38	[12, 136]	10.06	1.86
MSG class-1	49.43	3.66	88.20	[5, 150]	42.22	14.33
MSG class-2	33.67	3.64	61.93	[4, 100]	27.02	10.80
MSG class-3	379.96	66.10	24238.56	[4, 1000]	369.98	3.48
MSG class-4	332.00	4.00	664.00	[10, 1000]	377.22	25.97
MSG class-5	21.17	8.73	115.83	[12, 65]	11.80	1.93
MSG class-6	524.65	2.00	524.85	[49, 1000]	288.22	13.85

IMDB-B [16] (Social network). IMDB-B is a movie collaboration data set consisting of a self-network of 1,000 actors who play movie roles in IMDB. In each network, the nodes represent the actors/actresses. Researchers use an edge to link them if they act in the same movie. The criterion for classification is the type of movies. These networks are collected from the action movies and romantic movies.

MUTAG [17] (Small molecules). MUTAG is a data set of nitroaromatic compounds designed to predict their mutagenicity against salmonella typhimurium. The graphs are used to represent compounds, where nodes represent atoms and are labeled by atomic type (represented by single encoding), while edges between nodes represent bonds between corresponding atoms. It includes 188 compound samples and 7 discrete node labels.

In this article, we collect a number of empirical networks—including the set of protein structure data, macromolecular compound structure data, and social networks data—in combination with the three typical modeled networks of BA [37], WS [36] and ER [35] to create a synthesis cross-scale graph classification benchmark data set MSG. Table I presents the basic statistical properties of open data sets and MSG. The large standard deviation of the node distribution reflects the large difference in the size of the graphs, which reflects the goal of cross-scale graph classification. A visual comparison of structures with varying sizes in different classes are depicted in Fig. 4.

As evident from the Fig. 3, the maximum number of nodes in graphs in MSG is 1,000, and the minimum number of graph's nodes is 4. The difference is approximately 10^3, which meets the definition of the cross-scale graphs. MSG consists mainly of three peaks: The first peak consists of graphs of nodes between 0 and 200, representing small-scale networks such as small molecular compounds in the real world. The second peak consists of graphs with 500-600 nodes, representing medium-scale complex networks, such as macromolecular networks and brain networks in the real world. The third peak consists of graphs with 900-1000 nodes, representing large graphs, such as social networks in the real world.

2) Baseline: To answer $Q1$, we select nine advanced methods for comparison:

Set2set [54]. This work presents a read-process-write framework for unordered output data, and proposes an efficient training algorithm (Set2set), which searches for the best possible output sequence during training and prediction.

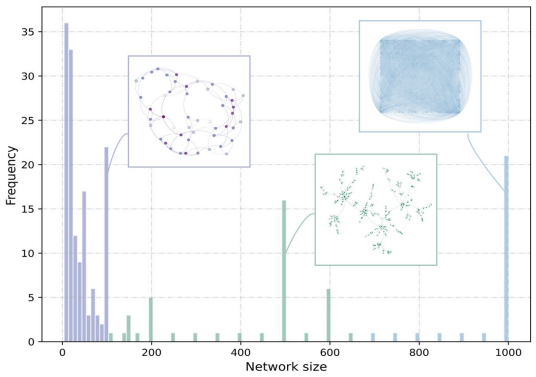


Fig. 3. The distribution of of the graph size of the MSG data set

GIN [43]. GIN is as powerful as the Weisfeiler-Lehman graph isomorphism test and it achieves state-of-the-art performance.

GMT [55]. GMT aims to enhance feature extraction capabilities through multiset operations and attention mechanisms. The core idea of GMT is to leverage the self-attention mechanism within the Transformer architecture to process graph data, thereby capturing the complex relationships between nodes and the overall structural information.

Diffpool [47]. Diffpool uses GNN models to learn a assign matrix which assigns a group of nodes into one node. Its pooling strategy makes the architecture end-to-end trainable. The node drop pooling uses a learnable scoring function to eliminate nodes with low scores. The researchers report that Diffpool has an advantage on big biology data sets in terms of accuracy.

Nested GCN [56]. Nested graph neural networks (Nested GNN) represents a graph with rooted subgraphs rather than rooted subtrees. Thus, the representations of two graphs that contain many identical subgraphs tend to be similar. It is reported that Nested GNN is highly competitive for graph classification tasks. We use GCN for its basic model.

DGCNN [57]. DGCNN uses WL algorithm [39] to generate features for nodes and propose a pooling method called SORTPOOL to select the first m nodes to create an equal-size graph which makes it convenient to use the CNN method to finish the graph classification task. Finally, a CNN is used for the graph classification tasks.

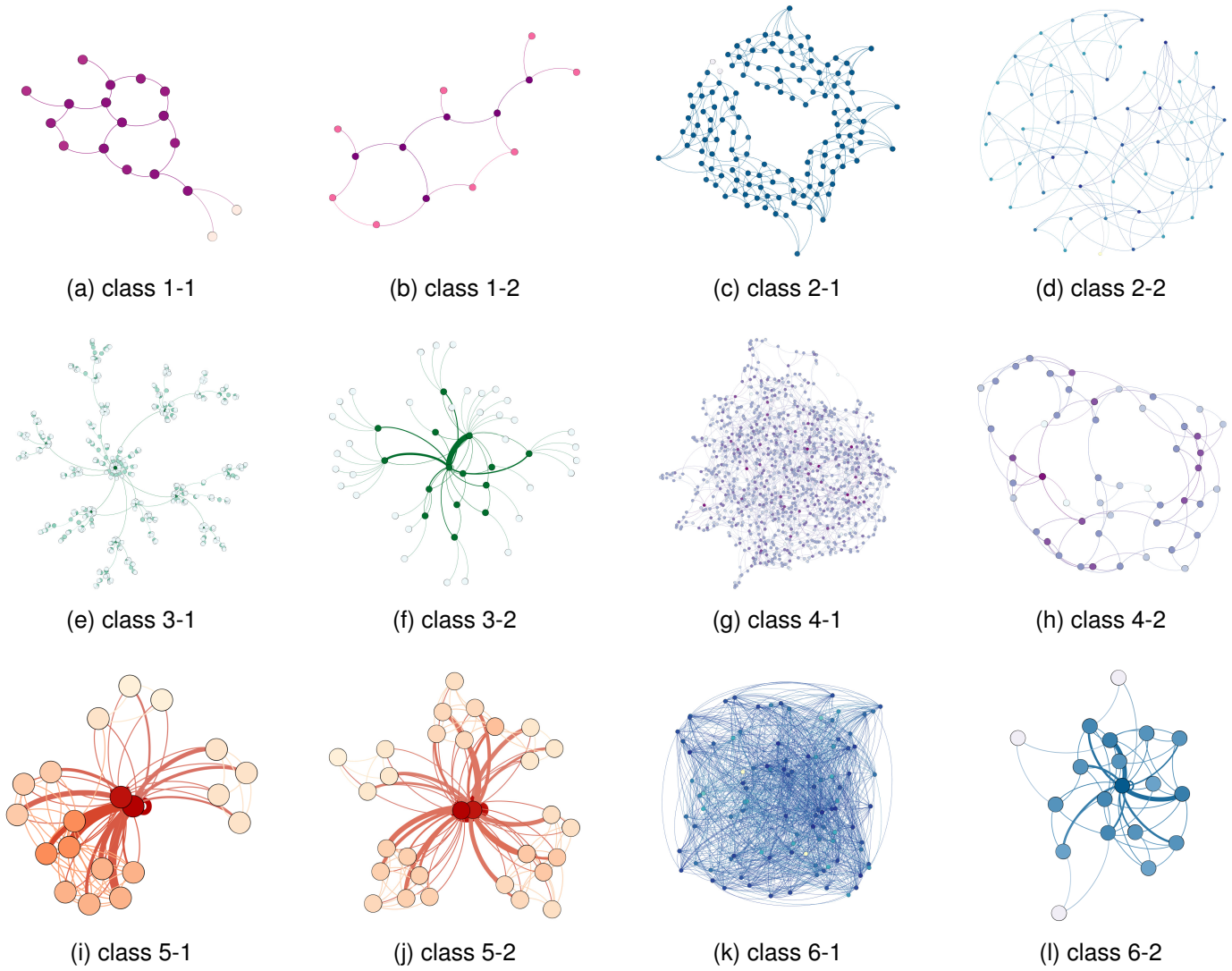


Fig. 4. Example of the MSG data set. class X-Y indicates that the graph is the Y-th example from class X.

G-Mixup [58]. G-Mixup uses random graph mixing to generate new graphs and, thus, to augment the original data set. Unlike traditional data enhancement methods, G-Mixup can be generated with different topologies. The graph effectively increases the diversity of the data sets. In addition, G-Mixup can also be used in combination with other data enhancement methods to further improve model performance.

ICL [59]. The Information-based Causal Learning (ICL) framework integrates information theory and causality to transform correlation into dependence. This model introduces a mutual information objective to enhance causal features rather than correlational patterns. The paper claims that ICL significantly improves accuracy and robustness in graph classification tasks.

GPN [60]. Graph Parsing Networks (GPNs) focus on parsing and comprehending complex graph structures. The aim of GPNs is to enhance the performance of graph representation and classification tasks by parsing the hierarchical structure of graphs and the relationships between nodes.

We utilise GCN [9] + Diffpool [47] for our ablation study's baseline model. We add wavelet convolution layer (GWC) and spectral-pooling respectively and test which part is more effective.

Our model and baseline models use the same network structure (for example, layers, activation functions) and the same training hyperparameters (for example, optimizer, learning rate, and gradient clipping). The proportion of training sets, test sets, and validation sets is 8:1:1.

B. Comparison between GSpect and Other Models

1) Classifying Graphs in Open Data Sets: The performance of GSpect and baseline models on the classification of open data sets are presented in Table II. GSpect achieves four of the best performance out of five data sets with the average improvements of 1.62% in classification accuracy. In particular, our model highly improves the performance (by 3.33%) for the biological macromolecules data set (PROTEINS). The reaon for this is that GWC captures multi-scale messages from a complex structure. Moreover, because every graph should be pooled into the same size, the spectral-pooling method saves most messages during the process of pooling on a larger

TABLE II
COMPARISON EXPERIMENT IN TERMS OF CLASSIFICATION ACCURACY BETWEEN GSPECT AND OTHER MODELS ON OPEN DATA SETS. THE BEST RESULTS ARE MARKED IN BOLD FONT AND SUB-BEST RESULTS ARE UNDERLINED.

Algorithm	PTC	MUTAG	PROTEINS	D&D	IMDB-B
Set2set	64.45 ± 5.51	71.90 ± 2.81	74.51 ± 2.26	76.42 ± 3.84	63.91 ± 4.10
GIN	64.13 ± 8.12	89.40 ± 5.6	76.46 ± 2.88	76.84 ± 3.11	74.66 ± 5.28
GMT	67.21 ± 1.32	88.24 ± 4.07	73.45 ± 6.02	78.41 ± 5.11	72.88 ± 2.61
Diffpool	66.65 ± 8.57	84.30 ± 2.56	$\underline{76.96 \pm 1.88}$	78.88 ± 2.87	65.61 ± 1.11
DGCNN	72.62 ± 1.76	84.66 ± 2.06	70.59 ± 0.34	$\underline{79.01 \pm 0.52}$	69.90 ± 0.29
NestedGCN	70.26 ± 4.18	73.81 ± 9.70	74.20 ± 2.50	76.53 ± 3.88	73.79 ± 1.18
GPN	74.68 ± 4.57	89.31 ± 3.10	76.22 ± 2.10	77.39 ± 4.69	73.50 ± 2.59
G-mixup	$\underline{74.41 \pm 1.62}$	87.98 ± 2.49	74.44 ± 1.63	78.61 ± 0.89	$\mathbf{83.84 \pm 3.20}$
ICL	73.02 ± 7.17	$\underline{89.57 \pm 4.06}$	75.21 ± 2.99	76.15 ± 2.56	74.59 ± 4.70
GSpect	$\mathbf{74.90 \pm 3.53}$	$\mathbf{91.11 \pm 4.68}$	$\mathbf{80.29 \pm 2.83}$	$\mathbf{80.14 \pm 4.38}$	$\underline{74.85 \pm 4.00}$

TABLE III
ABLATION STUDY BETWEEN GSPECT AND OTHER MODELS ON OPEN DATA SETS.

Algorithm	PTC	MUTAG	PROTEINS	D&D	IMDB-B	MSG
GCN+Diffpool	66.65 ± 8.57	84.30 ± 2.56	76.96 ± 1.88	77.88 ± 2.87	65.61 ± 1.11	57.14 ± 9.05
GCN+Spectral-pooling	67.06 ± 4.96	93.33 ± 5.11	81.18 ± 3.97	81.08 ± 7.02	74.10 ± 2.85	71.90 ± 8.73
GWC+Diffpool	70.83 ± 8.23	90.55 ± 3.75	78.56 ± 2.64	80.42 ± 3.45	71.86 ± 5.27	70.52 ± 6.48
GSpect	74.90 ± 3.53	91.11 ± 4.68	80.29 ± 2.83	80.14 ± 4.38	74.85 ± 4.00	75.33 ± 7.73

scale. It must be noted that at the data set IMDB-B, GSpect lags behind G-mixup. Table 1 indicates that the IMDB-B data set has a small average shortest path length, which implies that the distances between nodes are short (shaping a strong local structure), facilitating rapid information propagation. G-Mixup can effectively capture the local structures and relationships between nodes by generating new graphs through random graph mixing [58]. This augmented data set aids the model in better learning the interactions among nodes, thereby enhancing classification performance. Furthermore, G-Mixup's graph mixing strategy allows the model to generate diverse graph samples during training, which contributes to improving the model's generalization ability. In contrast, while GWC and spectral pooling demonstrate strong performance in aggregating multi-scale information, they may not adequately capture critical local structural information when dealing with social networks characterized by complex connectivity patterns. Additionally, the analysis in Appendix D indicates that there is a lower bound on the information loss during the pooling process of the spectral pooling layer, which also somewhat limits the classification accuracy of the method on a certain data set.

2) Classifying Graphs in Cross-scale Data Sets: The performance of GSpect and baseline models on the classification of MSG are presented in Fig. 5. We improved the average accuracy by 13.38% (average difference in accuracy between GSpect and all other methods). There are numerous reasons for this. First, the final GWC layer is composed of multi-scale GWC layers; thus, the advantage of GWC is its ability to capture the information of cross-scale structures in graphs. For the cross-scale graph data set MSG, GWC can better aggregate the structure information and generate graph representations. Second, because the graphs' adjacency matrix is usually different in size, the traditional methods are difficult to pool the graphs. However, these cross-scale graphs in the same class have a similar topology and also have a similar spectrum. Thus, the spectral-pooling method can accomplish the pooling task.

It must be pointed out that almost all methods yield a large standard deviation, which results in high uncertainty. This is due to a number of reasons. First, collecting cross-scale graph data is difficult and, thus, the sample space of MSG is small (210 samples), thereby leading to large fluctuations. Second, the large variation in graph size (almost 10^3) leads to the difficulty of classification, which results in a few wrong classification results.

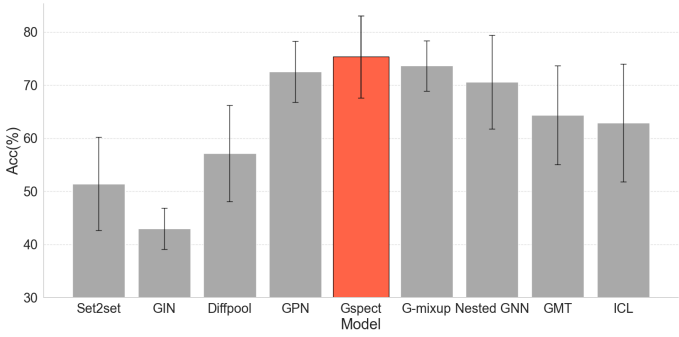


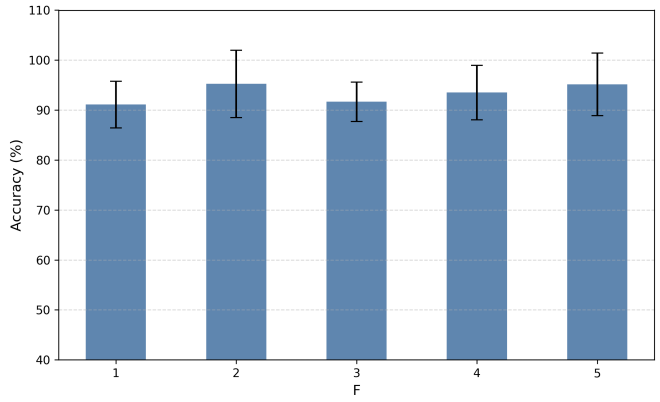
Fig. 5. Comparison experiment between GSpect and other models on MSG. The error bars represent the S.D..

C. Ablation Study

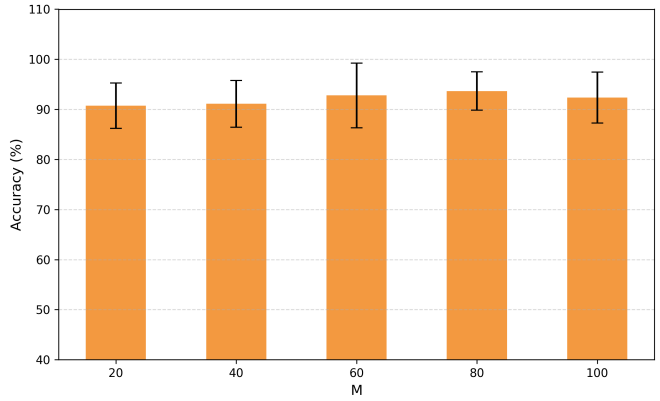
To answer $Q2$, we design an ablation study to verify which part of GSpect is significant and why GSpect has better performance.

Table III reports the results of ablation study. It is evident that GWC and the spectral-pooling layer improves the performance partly, which proves the effectiveness of GWC and spectral-pooling.

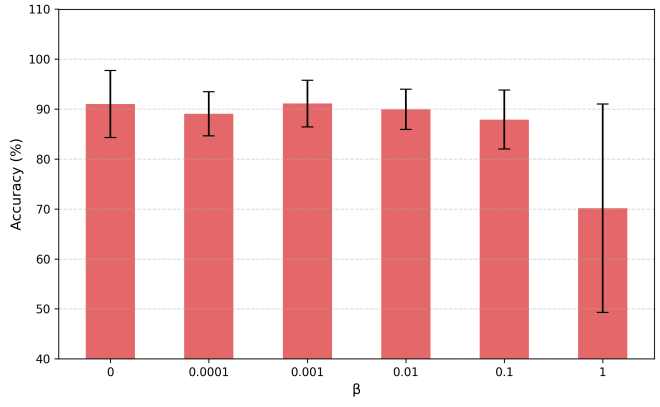
Note that using spectral pooling with the data sets MUTAG and PROTEINS, using spectral-pooling only leads to better performance than GSpect. The reason for this is that the GWC aggregates the multi-scale spectral messages as its output and



(a) Sensitivity analysis of F. The error bars represent the S.D..



(b) Sensitivity analysis of M. The error bars represent the S.D..



(c) Sensitivity analysis of β. The error bars represent the S.D..

Fig. 6. The results of sensitivity analysis.

the spectral-pooling layer filters the redundant messages and generates the principal component representation. However, in this study, we retain the first F-scale wavelet and loss portion of the high-scale messages. For MUTAG and PROTEINS, this method affects the accuracy of classification. Another reason is, unlike most other methods, GWC trains a non-sparse parameter matrix, which may lead to overfitting on these datasets. After removing GWC, the model complexity is reduced, which in turn improves its generalization ability.

With regard to stability, GWC and spectral-pooling partially increase the standard deviation of classification accuracy, which reduces the stability of the model. This is because these two methods have more learnable parameters which increase the difficulty of optimization and increase the probability of falling into local optimum.

D. Sensitivity Analysis

To answer $Q3$, we change the value of hyperparameters and observe the performance of GSpect in the classification accuracy. The experiment is based on MUTAG. Fig. 6 presents the results of the sensitivity analysis. According to Fig. 6, we find that GSpect undergoes small changes when the number of Chebyshev polynomials M and the number of wavelet scale F changes. This result implies that on the basis of maintaining high classification accuracy, researchers can select small F and M to reach lower code execution time.

Notably, the accuracy is 91.11% when $\beta = 0.001$, whereas when $\beta = 0$, the accuracy is 91.01%, which is lower than that at $\beta = 0.001$. Additionally, in terms of standard deviation, results at $\beta = 0$ has a standard deviation of 6.71%, compared to 4.68% when $\beta = 0.001$. This suggests that the term contributes to enhancing model stability, consistent with the analysis presented in the Appendix A. However, this improvement is limited (from 91.01% to 91.11%). Future studies can further evaluate the effect of L_p in various tasks. Besides, the classification accuracy reduces sharply when the equilibrium coefficient β increases. As Equation 22 shows, when β is close to 1, L_p plays a leading role in the optimization function. The results reveal that using L_p alone will reduce the performance of GSpect. Thus, researchers need to adjust β to ensure that the two optimization function have the same order of magnitude.

V. Conclusion

Considering there is few methods for cross-scale graph classification tasks, we proposed GSpect, an advanced cross-scale graph classification model in this study. We use the graph wavelet neural network as the convolution layer which improved the performance of obtaining graph-level representations. In addition, we designed the spectral-pooling layer which filters useless messages directly on the spectrum and aggregates the nodes to resize the graph by spectral pooling. Based on the fact that there is few cross-scale graph data sets, we collect data and create the cross-scale data set MSG. We compared this data set with the state-of-the-art ones to prove the superiority of the classification accuracy of GSpect using both open data sets and MSG. Experiments reveal that, on open data sets, GSpect improves the performance of classification accuracy by 1.62% on average, and for a maximum improvement of 3.33% on PROTEINS. On MSG, GSpect improves the performance of classification accuracy by 13.38% on average. Further, we employed an ablation study to observe the improve of accuracy by GWC and spectral-pooling. The results reveal that when we employed them simultaneously, we obtain the best results with regard to graph

classification, which proves that it is necessary to use them simultaneously. Further, we conducted the sensitivity analysis to verify the stability of GSpect when there is a change in the hyperparameters. The results reveal that researchers can select a small F and M but need to decide the value of β carefully. While GSpect demonstrates excellent performance in cross-scale graph classification tasks, we acknowledge certain limitations inherent in our approach. Notably, the spectral filtering based on K-order polynomials effectively covers only the K-order subgraphs of the central node, suggesting that this method may, in itself, represent a suboptimal solution [61]. Future research directions could explore methods to overcome this limitation, such as integrating complementary techniques to capture more comprehensive graph structural information, or developing novel spectral filtering approaches that more effectively address long-range dependencies. Besides, we candidly acknowledge that the model's computational complexity is indeed high ($O(n^3 + Fn^2l)$, see in Appendix B), due to the wavelet transform methods used, which require calculating the pseudoinverse and performing Fourier transforms. However, this high computational complexity results in higher classification accuracy. Additionally, as mentioned in the sensitivity analysis section, adjusting the hyperparameters M and F can reduce computational complexity while maintaining high classification accuracy. Future work can focus on reducing the increased computational complexity associated with wavelet-based methods. This work fills the gap of lacking cross-scale graph classification research. Besides, GSpect fills the gap in extant literature regarding a lack of cross-scale graph classification studies and could facilitate application research, for example, predicting the function of protein in accordance with its structure and enabling the selection of appropriate drugs.

REFERENCES

[1] D. Whitford, *Proteins: structure and function*. John Wiley & Sons, 2013.

[2] X. Lu, D. J. Wrathall, P. R. Sundsøy, M. Nadiruzzaman, E. Wetter, A. Iqbal, T. Qureshi, A. J. Tatem, G. S. Canright, K. Engø-Monsen *et al.*, "Detecting climate adaptation with mobile network data in bangladesh: Anomalies in communication, mobility and consumption patterns during cyclone mahasen," *Climatic Change*, vol. 138, pp. 505–519, 2016.

[3] E. E. Schadt, S. H. Friend, and D. A. Shaywitz, "A network view of disease and compound screening," *Nature Reviews Drug Discovery*, vol. 8, no. 4, pp. 286–295, 2009.

[4] B. Wu, C. Yuan, and W. Hu, "Human action recognition based on context-dependent graph kernels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2609–2616.

[5] J. B. Lee, X. Kong, C. M. Moore, and N. K. Ahmed, "Deep parametric model for discovering group-cohesive functional brain regions," in *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 2020, pp. 631–639.

[6] N. Brown, "Chemoinformatics—an introduction for computer scientists," *ACM Computing Surveys (CSUR)*, vol. 41, no. 2, pp. 1–38, 2009.

[7] J. Wen, E. Thibeau-Sutre, M. Diaz-Melo, J. Samper-González, A. Routier, S. Bottani, D. Dormont, S. Durrleman, N. Burgos, O. Colliot *et al.*, "Convolutional neural networks for classification of alzheimer's disease: Overview and reproducible evaluation," *Medical Image Analysis*, vol. 63, p. 101694, 2020.

[8] G. Nikolentzos, G. Siglidis, and M. Vazirgiannis, "Graph kernels: A survey," *Journal of Artificial Intelligence Research*, vol. 72, pp. 943–1027, 2021.

[9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[10] C. Zhuang and Q. Ma, "Dual graph convolutional networks for graph-based semi-supervised classification," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 499–508.

[11] S. Zheng, Z. Zhu, Z. Liu, Y. Li, and Y. Zhao, "Node-oriented spectral filtering for graph neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[12] J. Guo, K. Huang, X. Yi, and R. Zhang, "Graph neural networks with diverse spectral filtering," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 306–316.

[13] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 260–291, 2016.

[14] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2020.

[15] P. D. Dobson and A. J. Doig, "Distinguishing enzyme structures from non-enzymes without alignments," *Journal of Molecular Biology*, vol. 330, no. 4, pp. 771–783, 2003.

[16] P. Yanardag and S. Vishwanathan, "Deep graph kernels," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1365–1374.

[17] A. K. Debnath, R. L. Lopez de Compadre, G. Debnath, A. J. Shusterman, and C. Hansch, "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity," *Journal of Medicinal Chemistry*, vol. 34, no. 2, pp. 786–797, 1991.

[18] D.-A. Silva, S. Yu, U. Y. Ulge, J. B. Spangler, K. M. Jude, C. Labão-Almeida, L. R. Ali, A. Quijano-Rubio, M. Ruterbusch, I. Leung *et al.*, "De novo design of potent and selective mimics of il-2 and il-15," *Nature*, vol. 565, no. 7738, pp. 186–191, 2019.

[19] L. Cao, I. Goreshnik, B. Coventry, J. B. Case, L. Miller, L. Kozodoy, R. E. Chen, L. Carter, A. C. Walls, Y.-J. Park *et al.*, "De novo design of picomolar sars-cov-2 miniprotein inhibitors," *Science*, vol. 370, no. 6515, pp. 426–431, 2020.

[20] A. A. Glasgow, Y.-M. Huang, D. J. Mandell, M. Thompson, R. Ritterson, A. L. Loshbaugh, J. Pellegrino, C. Krivacic, R. A. Pache, K. A. Barlow *et al.*, "Computational design of a modular protein sense-response system," *Science*, vol. 366, no. 6468, pp. 1024–1028, 2019.

[21] Y. Hsia, J. B. Bale, S. Gonen, D. Shi, W. Sheffler, K. K. Fong, U. Nattermann, C. Xu, P.-S. Huang, R. Ravichandran *et al.*, "Design of a hyperstable 60-subunit protein icosahedron," *Nature*, vol. 535, no. 7610, pp. 136–139, 2016.

[22] O. C. Redfern, B. Dessailly, and C. A. Orengo, "Exploring the structure and function paradigm," *Current Opinion in Structural Biology*, vol. 18, no. 3, pp. 394–402, 2008.

[23] K.-L. Du, "Clustering: A neural network approach," *Neural Networks*, vol. 23, no. 1, pp. 89–107, 2010.

[24] S.-D. Tan, "A general s-domain hierarchical network reduction algorithm," in *ICCAD-2003. International Conference on Computer Aided Design (IEEE Cat. No. 03CH37486)*. IEEE, 2003, pp. 650–657.

[25] G. M. Slota, K. Madduri, and S. Rajamanickam, "Complex network partitioning using label propagation," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, pp. S620–S645, 2016.

[26] Z. Liu, Q. Mao, C. Liu, Y. Fang, and J. Sun, "On size-oriented long-tailed graph classification of graph neural networks," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1506–1516.

[27] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[28] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[29] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[30] N. Tremblay and P. Borgnat, "Graph wavelets for multiscale community mining," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5227–5239, 2014.

[31] B. Xu, H. Shen, Q. Cao, Y. Qiu, and X. Cheng, "Graph wavelet neural network," *arXiv preprint arXiv:1904.07785*, 2019.

[32] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Applied and computational harmonic analysis*, vol. 21, no. 1, pp. 53–94, 2006.

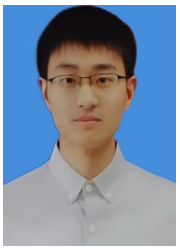
[33] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1320–1329.
[34] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.
[35] P. Erdős, A. Rényi *et al.*, "On the evolution of random graphs," *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, vol. 5, no. 1, pp. 17–60, 1960.
[36] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
[37] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
[38] G. Ma, N. K. Ahmed, T. L. Willke, and P. S. Yu, "Deep graph similarity learning: A survey," *Data Mining and Knowledge Discovery*, vol. 35, pp. 688–725, 2021.
[39] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels." *Journal of Machine Learning Research*, vol. 12, no. 9, 2011.
[40] R. Al-Rfou, B. Perozzi, and D. Zelle, "Ddgk: Learning graph representations for deep divergence graph kernels," in *The World Wide Web Conference*, 2019, pp. 37–48.
[41] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1263–1272.
[42] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.
[43] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?" *arXiv preprint arXiv:1810.00826*, 2018.
[44] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," *Advances in Neural Information Processing Systems*, vol. 28, 2015.
[45] Y. Ma, S. Wang, C. C. Aggarwal, and J. Tang, "Graph convolutional networks with eigenpooling," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 723–731.
[46] F. M. Bianchi, D. Grattarola, and C. Alippi, "Spectral clustering with graph neural networks for graph pooling," in *International Conference on Machine Learning*. PMLR, 2020, pp. 874–883.
[47] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
[48] S. Yahia, S. Said, and M. Zaied, "Wavelet extreme learning machine and deep learning for data classification," *Neurocomputing*, vol. 470, pp. 280–289, 2022.
[49] M. Behmanesh, P. Adibi, S. M. S. Ehsani, and J. Chanussot, "Geometric multimodal deep learning with multiscaled graph wavelet convolutional network," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
[50] G. B. Arfken, H. J. Weber, and F. E. Harris, *Mathematical methods for physicists: a comprehensive guide*. Academic Press, 2011.
[51] C. Liu, Y. Zhan, C. Li, B. Du, J. Wu, W. Hu, T. Liu, and D. Tao, "Graph pooling for graph neural networks: Progress, challenges, and opportunities," *arXiv preprint arXiv:2204.07321*, 2022.
[52] H. Toivonen, A. Srinivasan, R. D. King, S. Kramer, and C. Helma, "Statistical evaluation of the predictive toxicology challenge 2000–2001," *Bioinformatics*, vol. 19, no. 10, pp. 1183–1193, 2003.
[53] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, no. suppl_1, pp. i47–i56, 2005.
[54] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," *arXiv preprint arXiv:1511.06391*, 2015.
[55] J. Baek, M. Kang, and S. J. Hwang, "Accurate learning of graph representations with graph multiset pooling," *arXiv preprint arXiv:2102.11533*, 2021.
[56] M. Zhang and P. Li, "Nested graph neural networks," *arXiv preprint arXiv:2110.13197*, 2021.
[57] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
[58] X. Han, Z. Jiang, N. Liu, and X. Hu, "G-mixup: Graph data augmentation for graph classification," in *International Conference on Machine Learning*. PMLR, 2022, pp. 8230–8248.
[59] Z. Zhao, P. Wang, H. Wen, Y. Zhang, Z. Zhou, and Y. Wang, "A twist for graph classification: Optimizing causal information flow in graph neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 15, pp. 17 042–17 050, Mar. 2024. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/29648
[60] Y. Song, S. Huang, X. Wang, C. Zhou, and Z. Lin, "Graph parsing networks," *arXiv preprint arXiv:2402.14393*, 2024.
[61] J. Guo, K. Huang, X. Yi, Z. Su, and R. Zhang, "Rethinking spectral graph neural networks with spatially adaptive filtering," *arXiv preprint arXiv:2401.09071*, 2024.

Xiaoyu Zhang received the B.E. degree in automation from the National University of Defense Technology, Changsha, China. He is currently a PhD student at the College of Systems Engineering, National University of Defense Technology. His research interests include complex systems, physical networks and graph neural networks.

Wenchuan Yang received the B.E. degree in management science from the Sichuan University, Chengdu, China in 2019. He is currently a PhD student at the College of Systems Engineering, National University of Defense Technology. His research interests include recommender systems, complex networks, graph neural networks and data mining.

Jiawei Feng received the B.E. degree from Northeastern University, Shenyang, China. He is currently working to the M.S. degree in Management Science and Engineering at National University of Defense Technology. His research interests include complex system, supply chain management and data mining.

Bitao Dai is currently a PhD student at the College of Systems Engineering, National University of Defense Technology. He received his B.E. degree in School of Economics and Management from Southwest University. His research interests include social network analysis, network disintegration, human behavior and information diffusion.



Tianci Bu obtained the B.E. degree in computer science from Xi'an University of Architecture and Technology. He is currently a master student at the College of Systems Engineering, National University of Defense Technology. His research interests include natural language processing and big data analytics.

Xin Lu obtained his PhD from Department of Public Health Sciences at Karolinska Institutet, Stockholm, Sweden. He is the co-founder of Flowminder Foundation, and professor at the National University of Defense Technology in China. He has been granted the National Science Fund for Distinguished Young Scholars. His research interests include big data analytics, complex networks, human behavior and emergency management.