Revisiting Graph Wavelet Neural Network: Mis-claim and Universal Enhancement Framework

Guoming Li [1 2] **Jian Yang** [2] **Shangsong Liang** [3] **Dongsheng Luo** [4]

Abstract

Recently, an ICLR19' paper (Xu et al., 2019) has attracted intensive attention for proposing a new Graph Neural Network (GNN) model named Graph Wavelet Neural Network (GWNN). In that paper, they define the graph convolution with Graph Wavelet Transform (shortly GWConv) by trivially involving the Graph Wavelet Transform (GWT) to graph convolution, and further propose the GWNN based on the definition. However, the underlying rationality of this definition is a mis-claim due to the non-existence of the Fourier-typed convolution theorem in wavelet transform, making the proposed GWNN an inappropriate GNN implementation. In this work, we revisit the mis-claim of their definition, propose the correct formulation of GWConv, and prove the effectiveness and necessity of our proposition from the perspective of Graph Signal Processing (GSP). Based on the correct formulation, we propose Multi-Resolution Wavelet Enhancement, a universal framework that improves GNN performance by involving a generalized multi-resolution GWConv. Extensive experiments on benchmark datasets show that our framework can significantly and consistently improve the performances of both spatial-based and spectral-based GNNs on the node classification task.

1. Introduction

Signals generated from complex networks (Newman, 2010) and accompanied by complex topologies are called *graph signals* (Shuman et al., 2013). Such signals bring great challenges for signal analysis since traditional methods dismiss the latent structures of the signals. To overcome the limitations of traditional methods, Graph Signal Processing (GSP) (Sandryhaila & Moura, 2013b; Shuman et al., 2013) has been proposed to analyze the signals with consideration of topologies in graph signals. The studies of GSP are based on Graph Fourier Transform (GFT) (Sandryhaila & Moura, 2013a), a graph-based generalization of conventional Fourier Transform (FT) (Proakis, 1975). GFT bridges the gap between the vertex domain and frequency domain with a graph Laplacian (Chung, 1997), and further defines some important concepts, such as graph convolution (Gama et al., 2020b). All these have led to a better realization of the signal processing on graphs from the perspective of frequency (Ortega et al., 2018) and further achieved a lot of impressive results (Sharpnack et al., 2016; Drayer & Routtenberg, 2018; Cheung et al., 2018).

As the successful generalization from conventional FT to GFT, researchers have tried to generalize more techniques in conventional signal processing to GSP and obtained fruitful outcomes. One important result is Graph Wavelet Transform (GWT) (Hammond et al., 2011), which generalizes Wavelet Transform (WT) (Rioul & Vetterli, 1991) to signals on graphs and has achieved great success in various tasks (Tremblay & Borgnat, 2014; Chedemail et al., 2022).

Another important technique for GSP is Graph Neural Network (GNN). As the dominating deep learning framework in the graph structure domain, GNN can also be seen as a GSP model (Fu et al., 2020; Gama et al., 2020a). Existing GNN methods can be divided into two classes, spatial-based (Veličković et al., 2018; Kipf & Welling, 2017) and spectral-based (Bruna et al., 2014; Defferrard et al., 2016). Spatial-based methods that follow the principles of message-passing neural network (MPNN) (Gilmer et al., 2017) have achieved remarkable performance but leave the frequency of graph signals unexplored. In contrast, spectral-based methods process graph data based on GSP theory, making these methods theoretically well-grounded (Wu et al., 2021).

More recently, some efforts have been done to combine the advantages of GWT and GNN, represented by an ICLR'19 paper (Xu et al., 2019). That paper defined the *graph convolution with GWT* (shortly GWConv) by directly replacing GFT with GWT in graph convolution. Based on their

[1]Independent Researcher [2]Institute of Automation, Chinese Academy of Sciences [3]Mohamed bin Zayed University of Artificial Intelligence [4]Knight Foundation School of Computing and Information Sciences, Florida International University. Correspondence to: Guoming Li <PaskardLi@outlook.com>, Dongsheng Luo <dluo@fiu.edu>.

GWConv, they further propose Graph Wavelet Neural Network (GWNN), which is a GNN implementation for their GWConv. Unfortunately, the rationality of such direct replacement is a mis-claim due to the non-existence of the Fourier-typed convolution theorem in WT (R., 1994). Therefore, it is necessary to look for the correct formulation and appropriate GNN implementation for GWConv.

To address the problems above, in this work, we revisit the definition of GWConv in the GWNN paper, analyze the mis-claim of their definition from the perspective of GSP, and propose the correct formulation of GWConv based on the definition of graph convolution (Shi & Moura, 2019). With rigorous derivation and computational experiments, we validate the effectiveness of our proposition and show a filter-wise disjoint range of GWNN's proposition and ours, indicating the necessity of our proposition. Furthermore, based on the derived GWConv, we extend it to a universal framework, Multi-Resolution Wavelet Enhancement. This framework generalizes the linear GWConv to GNN-based and preserves the advantages of *multi-resolution analysis* (Unser & Blu, 2003; Dai et al., 2003), which is the essence of GWT. Extensive experiments show that we can significantly improve the performance of various GNNs when we combine them with the proposed framework, indicating the versatility and effectiveness of the proposed method.

We summarize our main contributions as follows.

1) We prove the mis-claim in GWNN and propose the correct formulation of GWConv via rigorous derivation. With solid computational experiments on both synthetic and real-world datasets, we show how our proposition properly approximates the graph convolution of two graph signals, while GWNN's deviates it substantially.
2) We generalize linear GWConv to GNN-based GWConv, and further propose a novel Multi-Resolution Wavelet Enhancement, a plug-and-use graph data augmentation-like method that preserves the key multi-resolution analysis in GWT.
3) Comprehensive studies demonstrate that the Enhancement framework can sufficiently boost the performances for both spatial-based and spectral-based GNNs on five benchmark datasets.

2. Notation and Preliminary

2.1. Notation

We use a small case letter above with an arrow, such as $\vec{v}$, to denote a *Vector* and $\vec{v}(i)$ to denote its i-th element. Similarly, we use a bold capital letter, e.g., $\mathbf{M}$ to denote a *Matrix* and $\mathbf{M}_{ij}$ to denote its element in the i-th row and the j-th column. We use $\langle \vec{u}, \vec{v} \rangle$ to denote inner-product of two vectors $\vec{u}$ and $\vec{v}$, and use $\vec{u} \odot \vec{v}$ to denote *Hadamard product* (Strang, 2006) of $\vec{u}$ and $\vec{v}$. We use $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ to describe an undirected graph, where $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, and $\mathbf{A} \in \{0,1\}^{N \times N}$ is the adjacency matrix. $\mathbf{A}_{ij} = 1$ for $(v_i, v_j) \in \mathcal{E}$, $\mathbf{A}_{ij} = 0$, otherwise. For each node pair (v_i, v_j), $\mathbf{A}_{ij} = \mathbf{A}_{ji}$. $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ denotes normalized graph Laplacian matrix of $\mathcal{G}$, where $\mathbf{D} \in \mathbb{R}^{N \times N} = diag(d_{1,1}, d_{2,2}, ..., d_{N,N})$ denotes the diagonal degree matrix with $d_{i,i} = \sum_{i=1}^{N} \mathbf{A}_{ij}$, for $i = 1, 2, ..., N$. The $\vec{x} *_{\mathcal{G}} \vec{y}$ denotes a graph convolution operation of two graph signals $\vec{x}$, $\vec{y}$ on graph $\mathcal{G}$, no matter what transform it used. Notably, in this paper, we assume elements of both graphs and graph signals are **real** numbers which is a widely-accepted setting in graph learning (Hamilton, 2020; Gadde et al., 2014; Gao et al., 2019).

2.2. Graph Convolution in Vertex Domain

There are two types of graph convolution: *frequency-wise* and *vertex-wise* in the studies of GSP (Shi & Moura, 2019). In this paper, we focus on the more common vertex-wise graph convolution, which can be defined from two different aspects as follows.

Graph Convolution with Localized Graph Filters. Similar to the conventional definition of filters in digital signal processing, (localized-) graph filters are defined as systems that take a graph signal as input and produce another graph signal as output (Shuman et al., 2013; Sandryhaila & Moura, 2013c). Formally, given a graph signal $\vec{x}$ on a graph $\mathcal{G}$, the (localized-) graph filter and graph convolution on $\vec{x}$ can be written as:

$$\begin{aligned} \text{Graph Filter}: \quad & \mathbf{H} = \sum_{k=0}^{K} h_k \mathbf{S}^k \\ \text{Graph Convolution}: \quad & \mathbf{H}\vec{x} = \sum_{k=0}^{K} h_k \mathbf{S}^k \vec{x}. \end{aligned} \tag{1}$$

Here, the matrix $\mathbf{S}$ with weight $\{h_0, h_1, ..., h_K\}$ denotes *Graph Shift* which is associated with graph and satisfies $\mathbf{S}_{ij} = 0$ if $(v_i, v_j) \notin \mathcal{E}$ for $j \neq i$. As proved in (Duncan, 2004), graph shift with order k brings k-hops neighbors at each node into constructing the filter and computing the convolution. This property makes both filter and convolution "localized" in graphs.

Graph Convolution as Convolution of two Graph Signals. Akin to the classical convolution operation on two signals in the time domain, graph convolution can also be defined as convolution operation on two graph signals. Different from other kinds of convolution operations such as *Circular Convolution* and *Linear Convolution* (Proakis, 1975), graph convolution is determined by not only graph signals but also the topological structure (Shi & Moura, 2019). Specifically, the convolution of two graph signals $\vec{x}, \vec{y}$ on a given graph

$\mathcal{G}$ can be computed as follows.

$$\vec{x} *_{\mathcal{G}} \vec{y} = (\mathbf{U}^T)^{-1}(\mathbf{U}^T\vec{y} \odot \mathbf{U}^T\vec{x}) . \quad (2)$$

Here, the orthogonal matrix $\mathbf{U}^T$ denotes GFT operator of graph $\mathcal{G}$, which has two types of definitions (Zhu & Rabbat, 2012; Sandryhaila & Moura, 2013a):

$$\begin{aligned} \text{Laplacian-based}: &\quad \mathbf{L} = \mathbf{U}_L\mathbf{J}_L(\mathbf{U}_L)^{-1} \\ \text{Adjacency-based}: &\quad \mathbf{A} = \mathbf{U}_A\mathbf{J}_A(\mathbf{U}_A)^{-1}. \end{aligned} \quad (3)$$

The Jordan normal form $\mathbf{J}_L$ ($\mathbf{J}_A$) here is a diagonal matrix whose diagonal entries are eigenvalues of $\mathbf{L}$ ($\mathbf{A}$). Correspondingly, the row vectors of $\mathbf{U}_L^T$ ($\mathbf{U}_A^T$) are orthogonal eigenvectors of $\mathbf{L}$ ($\mathbf{A}$). Please note that we use *Laplacian-based* (specifically, the *normalized Laplacian*) in this paper.

With Eq. (3), the GFT of graph signal $\vec{x}$ and inverse GFT of the corresponding $\vec{\hat{x}}$ in frequency domain are defined as:

$$\begin{aligned} &\vec{\hat{x}} = \mathbf{U}^T\vec{x} \quad \text{(Graph Fourier Transform)} \\ &\vec{x} = \mathbf{U}\vec{\hat{x}} \quad \text{(Inverse Graph Fourier Transform)}. \end{aligned} \quad (4)$$

2.3. Wavelet Transform and Graph Wavelet Transform

Wavelet Transform. *Wavelets* are functions that satisfy certain mathematical requirements and often used to represent data or other functions (Graps, 1995; Rioul & Vetterli, 1991). As one of the most well-studied applications of wavelet theory, *Wavelet Transform* (WT) possesses strong capabilities in decomposing and analyzing highly complex signals, while those signals usually bring great troubles to conventional *Fourier Transform* (FT) (Unser & Blu, 2003). Unlike the Fourier basis used in FT, WT applies a special *wavelet basis* $w(t) = g(t)e^{2j\pi(f_0 s)\cdot t}$ (Rioul & Vetterli, 1991) with several important components: f_0 denotes basic frequency, which leads to a concept named "mother wavelet" (Graps, 1995); $g(t)$ is a window (band-pass) function, which is similar to the same term in windowed FT (Allen & Rabiner, 1977); s denotes *scale*, which controls the size of the window function and leads to a variable wavelet basis. The conventional WT of a signal $x(t)$ is shown as follows:

$$CWT_x(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t)\overline{w}(\frac{t-\tau}{s})dt , \quad (5)$$

where $\overline{w}(\cdot)$ is the conjugate of $w(\cdot)$, and τ is time location.

Graph Wavelet Transform. As a successful generalization of WT, *Graph Wavelet Transform* (GWT) (Hammond et al., 2011) of a graph signal $\vec{x}$ is defined as:

$$\vec{W_x}(s) = \sum_{l=1}^{N} g(s\lambda_l)\vec{\hat{x}}(l)\vec{u_l} . \quad (6)$$

Here, N denotes the number of nodes in the graph; s is the scale; $g(\cdot)$ denotes wavelet kernel function (related to window function $g(t)$ above); λ_l is an eigenvalue of graph Laplacian and $\vec{u_l}$ is the corresponding eigenvector; $\vec{\hat{x}}$ denotes GFT of graph signal $\vec{x}$ as mentioned in Eq. (4).

Furthermore, considering the $\vec{\hat{x}}$ is defined as:

$$\vec{\hat{x}} = (\vec{u}, \vec{u_2}, ..., \vec{u_N})^T\vec{x} = \mathbf{U}^T\vec{x} . \quad (7)$$

Eq. (6) can be further derived as:

$$\begin{aligned} \vec{W_x}(s) &= (\vec{u}, \vec{u_2}, ..., \vec{u_N}) \\ &\quad \cdot \begin{pmatrix} g(s\lambda_1) \\ g(s\lambda_2) \\ \vdots \\ g(s\lambda_N) \end{pmatrix} \odot \begin{pmatrix} \vec{\hat{x}}(1) \\ \vec{\hat{x}}(2) \\ \vdots \\ \vec{\hat{x}}(N) \end{pmatrix}) \\ &= \mathbf{U}diag[g(s\vec{\lambda})]\mathbf{U}^T\vec{x} \\ &= \psi_g^s\vec{x} . \end{aligned} \quad (8)$$

Here, ψ_g^s is known as *graph wavelet operator*, which can be efficiently approximated with polynomials (Hammond et al., 2011).

2.4. Spectral-based Graph Neural Networks

Spectral-based GNNs can be seen as GSP models which implement graph convolution with complex graph filters (Gama et al., 2020a). Existing works are constructed by either pre-defined (Gasteiger et al., 2019) or learnable graph filters (Defferrard et al., 2016; He et al., 2021), while both classes are usually approximated with polynomial functions as follows.

$$\mathbf{H} = \sum_{k=0}^{K} w_k T_k(\mathbf{S}), \quad \mathbf{S} = \mathbf{L} \text{ or } \mathbf{A} , \quad (9)$$

where $T_k(\cdot)$ denotes k order polynomial associated with a weight w_k for approximation.

Previous works have achieved promising results by designing powerful and expressive graph filters for spectral-based GNNs. For instance, ChebNet (Defferrard et al., 2016) uses the Chebyshev polynomial expansion of the eigenvalue-wise diagonal matrix to approximate the graph convolution with a significant reduction in computational complexity. ARMA (Bianchi et al., 2020) is a GNN implementation of autoregressive moving average graph filter (Isufi et al., 2017), which is a rational graph filter. BernNet (He et al., 2021) applies Bernstein polynomials to construct a non-negative graph filter and achieves state-of-the-art performance in various tasks.

3. GWNN revisited: Mis-claim and Correct Formulation

In this section, we focus on the mis-claim in the GWNN paper (Xu et al., 2019) and then propose the correct formu-

lation of GWConv. We further present rigorous derivations and conduct extensive computational experiments to verify the effectiveness and necessity of our proposition.

3.1. Mis-claim in GWNN

As proposed in (Xu et al., 2019), their GWConv was defined by applying GWT to the convolution of two graph signals $\vec{x}$ and $\vec{y}$ in an ad-hoc fashion. Specifically, their GWConv with diffusion kernel (Kondor & Lafferty, 2002) and scale s was obtained by simply replacing the GFT operator in Eq. (2) with the graph wavelet operator as follows:

$$\vec{x} \ *_{\mathcal{G}} \ \vec{y} = (\psi_g^s)^{-1}(\psi_g^s \vec{y} \odot \psi_g^s \vec{x}) \ . \tag{10}$$

However, an underlying assumption behind the replacement is that the convolution theorem of Fourier-typed holds for the GWT, making such an equation about GWConv as follows:

$$\psi_g^s(\vec{x} \ *_{\mathcal{G}} \ \vec{y}) = \psi_g^s \vec{y} \odot \psi_g^s \vec{x} \ . \tag{11}$$

However, as proved in (R., 1994), the convolution theorem of Fourier-typed does not hold for the conventional wavelet transform. Furthermore, such Fourier-typed convolution theorem does not hold for the GWT either because GWT is strictly derived from the conventional WT and faithfully preserves its computational properties (Hammond et al., 2011). Hence, GWConv cannot be rewritten as the element-wise product of their wavelet transform as Eq.(11), making the GWConv proposed in the GWNN paper a mis-claim.

3.2. Correct Formulation of GWConv

With Eq. (6)∼(8), we can therefore propose signal convolution such that $\vec{x} \ *_{\mathcal{G}} \ \vec{y}$ with GWT.

Proposition 3.1. *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, and two graph signals $\{\vec{x}, \vec{y}\}$, their GWConv $\vec{x} *_{\mathcal{G}} \vec{y}$ with (invertible-) wavelet kernel function $g(\cdot)$ and scale s is defined as:*

$$\vec{x} \ *_{\mathcal{G}} \ \vec{y} = (\psi_g^s)^{-1}\theta(\vec{y})\psi_g^s \vec{x} \ , \tag{12}$$

where $\theta(\vec{y})$ is a symmetric matrix determined by $\vec{y}$ with an expression as:

$$\theta(\vec{y}) = \mathbf{U} diag(\mathbf{U}^T \vec{y}) \mathbf{U}^T \ . \tag{13}$$

Proof. The proof can be found in Appendix A.1 □

3.3. Distance to "Ground-truth" Graph Convolution

To intuitively compare the proposition of GWNN with ours, in this section we empirically show how our proposition tightly matches the definitions of graph convolution in Sec 2.2 (both Eq. (1) and Eq. (2)), when GWNN's deviates from it substantially.

Table 1. Graph Convolution.

Transform	Math-Definition	Notation
Ours	Prop. 3.1	$\mathbf{M}_1$
GWNN	Eq. (10)	$\mathbf{M}_2$
GFT (Baseline)	Eq. (2)	$\mathbf{M}$

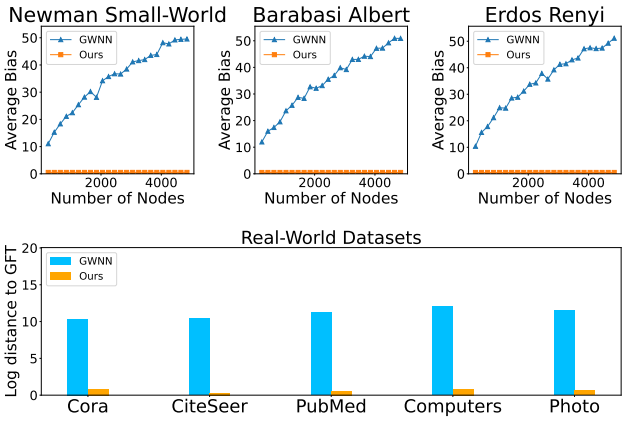

Figure 1. Graph Convolution Distance on both synthetic (Newman graphs, BA graphs, ER graphs) and five real-world Datasets. The blue denotes the results of GWNN and the orange is our method. Note that **higher** denotes the **worse** approximation to the GFT.

Our empirical proof is motivated by the uniqueness of graph convolution mentioned in Sec 2.2. Such uniqueness indicates that when both graph $\mathcal{G}$ and two graph signals $\vec{x}$, $\vec{y}$ are given, the convolution $\vec{x} \ *_{\mathcal{G}} \ \vec{y}$ can be properly computed by applying GFT as in Eq. (2). That is, the convolution result of two graph signals is determined once the signals and graph are determined. Thus, we can easily obtain the correct convolution result following Eq. (2), and evaluate the convolution results of other transforms with this "ground-truth". Specifically, we use three values $\mathbf{M}_1$, $\mathbf{M}_2$ and $\mathbf{M}$ to denote the convolution result using our method, GWNN's method, and the GFT, respectively (as shown in Table 1). Then we compute $||\mathbf{M}_i - \mathbf{M}||_F, i = 1, 2$ to measure the distance from the proposed GWConv to GFT. The smaller distance is better and denotes a correct convolution formulation.

We conduct comprehensive computation experiments on both synthetic and real-world datasets (See Appendix C for details of experimental settings). As shown in Figure 1, with scale $s = 1.5$, the convolution results of GWNN's GWConv deviate substantially from the baseline on all datasets, while our proposition has a tight approximation. More computational experiment results with different scales and networks on both synthetic datasets and real-world datasets can be found in Appendix D. The extensive experiments intuitively verify the correctness of our proposition and the inappropriateness of GWNN's.

3.4. Disjoint Range of Two Filters

Although the formula of the proposed GWConv (Eq. (12)) and the one in GWNN (Eq. (10)) look similar, they are completely different from the aspect of graph filters. In this section, we will theoretically prove a huge filter-wise gap between Eq. (10) and Eq. (12).

The Eq. (10) can be rewritten as follows:

$$(\psi_g^s)^{-1}(\psi_g^s\vec{y} \odot \psi_g^s\vec{x}) = (\psi_g^s)^{-1} diag(\psi_g^s\vec{y})\psi_g^s\vec{x}\,. \quad (14)$$

Notably, the only difference between the right side of Eq. (14) and the right side of Eq. (12) is the filter term, i.e., $\theta(\vec{y})$ and $diag(\psi_g^s\vec{y})$. Therefore, we theoretically investigate the gap between our GWConv and GWNN's from the perspective of graph filter. We first involve an auxiliary definition named *linear matrix span* as follows.

Definition 3.2. (Linear Matrix Span) Given a finite set of square matrices with the same order: $\mathcal{M} = \left\{\mathbf{M}_k \,:\, \mathbf{M}_k \in \mathbb{R}^{N\times N}, k = 1, 2, ...K\right\}$, the *Linear Matrix Span* (LMS) of $\mathcal{M}$ is a *matrix space* defined as:

$$LMS(\mathcal{M}) = \left\{\hat{\mathbf{M}} \,:\, \hat{\mathbf{M}} = \sum_{i=1}^{K} a_i\mathbf{M}_i, a_i \in \mathbb{R}\right\} \quad (15)$$

Notably, different from the proposition with a similar name in (Micheli et al., 2015), the LMS here is linear combinations rather than linear transformation. Moreover, LMS is equivalent to a $vector\ space$ if we implement *Vec operator* (V.Henderson & S.R.Searle, 1981) to each element in $LMS(\mathcal{M})$. Thus, we use this definition for intuitiveness and conciseness.

We further obtain two lemmas as follows:

Lemma 3.3. *Let $\vec{y}$ to be a N-dimension random vector that satisfies $\vec{y} \in \mathbb{R}^N$. The range of $\theta(\vec{y})$ is determined to be $LMS(\left\{\vec{u_i}\vec{u_i}^T : i = 1, 2, ..., N\right\})$. Similarly, when $g(s\lambda_i) \neq 0, i = 1, 2, ..., N$, the range of $diag(\psi_g^s\vec{y})$ is $LMS(\{diag(\vec{u_i}) : i = 1, 2, ..., N\})$.*

Proof. The proof can be found in Appendix A.2 □

Lemma 3.4. *The filter term $\theta(\vec{y})$ is a diagonal matrix **iff** $\vec{y} = t\sum_{i=1}^{N}\vec{u}_i$, where t is an arbitrary real number. In this case, the diagonal matrix $\theta(\vec{y})$ is $t \cdot \mathbf{I}$.*

Proof. The proof can be found in Appendix A.3 □

The $t \cdot \mathbf{I}$ here is an all-pass filter, which is often used as a phase or delay equalizer that just changes the phase but not the frequency components of the signal in conventional signal analysis (Mitra & Hirano, 1974; Regalia et al., 1988; Lang, 1998). However, both graph signals and graph filters (and their transforms in the frequency domain) are real-valued, which means the phase is ignored. As a result, the all-pass filter is not suitable and should be dismissed here. With this key insight and two lemmas: Lem. 3.3 and Lem. 3.4, we obtain a theoretical result about the disjoint range of two filters $\theta(\vec{y})$ and $diag(\psi_g^s\vec{y})$ as follows.

Proposition 3.5. *When ignoring the all-pass kernel mentioned in Lem. 3.4, the intersection of the range of two filters, $\theta(\vec{y})$ and $diag(\psi_g^s\vec{y})$, is empty.*

Proof. The proof can be found in Appendix A.4 □

This proposition explicitly indicates that the mis-claim keeps the GWConv in GWNN far away from the correct formulation we derived, which further demonstrates the necessity of our proposition.

4. Multi-Resolution Wavelet Enhancement

Based on the proposed GWConv, in this section, we present a novel method, Multi-Resolution Wavelet Enhancement. We will start with motivations and then introduce our enhancement framework.

4.1. Generalized and Expressive Graph Filters

As mentioned in Sec 2.2 (especially Eq. (1) and Eq. (2)), graph convolution operation can be interpreted as signals processing with corresponding graph filters (Yamada, 2022; Kenlay et al., 2021; Patane', 2022). Interestingly, there is a similar interpretation to GNN models. An insightful view on GNN is to regard it as a generalized graph filter that is designed for specific graph signal processing tasks (Ma et al., 2020; Fu et al., 2020; Gama et al., 2020a; Dong et al., 2020). That is, the data processing in GNN can be seen as a filtering operation on input graph signals with a complex (even non-analytic) graph filter. Such novel insight leads to a generalized definition of graph filters and makes it possible to design more expressive graph filters. Specifically, the linear filter $\theta(\vec{y})$, which denotes a correct formulation of graph filter in GWConv, can also be generalized with GNN. Therefore, we replace the linear filter $\theta(\vec{y})$ with a more complex and expressive GNN model, leading to a generalized GWConv as:

$$\theta(\vec{y})\psi_g^s\vec{x} \overset{\text{Generalized}}{\Longrightarrow} GNN(\psi_g^s\vec{x}, \mathcal{G}; \Omega). \quad (16)$$

Here, Ω denotes the (learnable) parameters of the GNN model.

4.2. Multi-Resolution Analysis (MRA)-based Graph Data Augmentation

MRA with Wavelet Transform. As mentioned in Sec. 2.3, WT performs much better than FT in processing complex

signals. The strong performance of WT owes a lot to a key technique named *Multi-Resolution Analysis (MRA)* (Chui & ao Lian, 1996; Dai et al., 2003). MRA applies an essential multiple scales mechanism that does not hold for FT and leads to a variable resolution for time-scale (equal to time-frequency) joint analysis. All these further lead to a more comprehensive investigation of the highly complex signals in real life (Mahmoodabadi et al., 2005; Ganesan et al., 2005). (See Appendix B for more details about MRA.)

Such capability also holds for GWT, making GWT outperforms GFT in GSP tasks (Hammond et al., 2011; Narang & Ortega, 2012; Tremblay & Borgnat, 2014). Different from GWNN, which only uses a single scale, we take the most important advantages of MRA into our proposed framework.

Graph Data Augmentation (GDA) and MRA-based GDA. We apply MRA to construct our framework following a technique named *Graph Data Augmentation* (GDA) (Zhao et al., 2022; Ding et al., 2022). As a data augmentation (DA) (Feng et al., 2021; Cubuk et al., 2019) technique for graph data, GDA has shown promising results for improving the performance of GNN models via augmenting graphs at a within-graph level (Rong et al., 2020) or between-graph level (Han et al., 2022; Verma et al., 2019). The essence of GDA is to improve GNN's performance on downstream tasks by enhancing the preserved information of original data. Specifically, GDA tries to find a projection $\rho(\cdot,\cdot)$ that generates augmented graph signal(s) $\mathbf{X}_i$ and graph(s) $\mathcal{G}_i$ with original data $(\mathbf{X},\mathcal{G})$ as follows (Ding et al., 2022):

$$\rho(\mathbf{X},\mathcal{G}) = \{(\mathbf{X}_1,\mathcal{G}_1),...,(\mathbf{X}_T,\mathcal{G}_T)\}\,. \tag{17}$$

Notably, the MRA mentioned above can be seen as a special GDA if we set a linear transformation ψ_g^s that **only** works on graph signal $\mathbf{X}$ (but not change graph $\mathcal{G}$). Formally:

$$\rho_{\text{MRA}}(\mathbf{X},\mathcal{G}) = \{(\psi_g^{s_1}\mathbf{X},\mathcal{G}),...,(\psi_g^{s_T}\mathbf{X},\mathcal{G})\}\,. \tag{18}$$

On top of this key insight, we further propose our Multi-Resolution Wavelet Enhancement as an MRA-based graph data augmentation, which will be intuitively shown in the following section.

4.3. Enhancement Framework

Based on the motivations above, our enhancement framework is designed to be combined with a specific GNN model defined as $\mathbf{Z} = f(\mathbf{X},\mathcal{G})$, where $\mathbf{Z}$, $\mathbf{X}$, $\mathcal{G}$, and f denotes node embeddings, graph signals (node features), graph structure, and signal transformation function of the GNN model, respectively. Note that the GNN model here is treated as the generalized graph filter that replaces the linear filter $\theta(\vec{y})$ as mentioned in Sec 4.1.

The overall framework is intuitively shown in Figure 2. Following Eq. (18), we first construct multi-resolution GWT

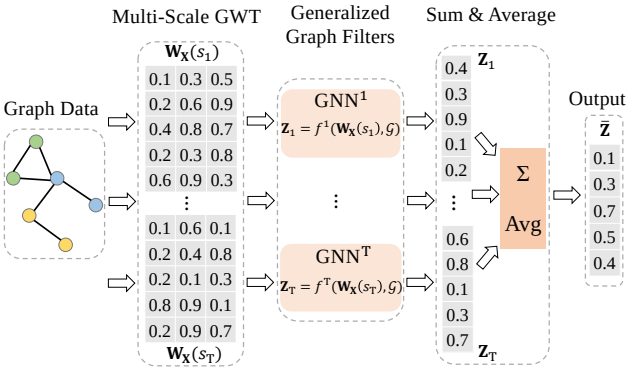

Figure 2. Overall Framework of Multi-Resolution Wavelet Enhancement.

with a diffusion kernel $g(\cdot)$ and a set of pre-defined scales $\{s_1, s_2, ..., s_T\}$ as follows.

$$\mathbf{W}_{\mathbf{X}}(s_i) = \psi_g^{s_i}\mathbf{X}, \quad i = 1,2,...,T\,. \tag{19}$$

Next, we take the constructed $\mathbf{W}_{\mathbf{X}}(s_i)$ as the input to the i-th GNN model f^i and then obtain the corresponding output $\mathbf{Z}_i$ as Eq. (20) shows:

$$\mathbf{Z}_i = f^i(\mathbf{W}_{\mathbf{X}}(s_i),\mathcal{G}), \quad i = 1,2,...,T\,. \tag{20}$$

Inspired by ARMA filter (Bianchi et al., 2020), the final output $\bar{\mathbf{Z}}$ is obtained by summing and averaging the $\mathbf{Z}_i$ as follows.

$$\bar{\mathbf{Z}} = \frac{\sum_{i=1}^{T}\mathbf{Z}_i}{T}\,. \tag{21}$$

Compared to GWNN, we dismiss the inverse operator for two reasons. First, since most graph learning tasks focus on dimension reduction, the output of GNN always keeps a much lower feature dimension than the input. Thus, it is unreasonable to apply the inverse GWT operator $(\psi_g^s)^{-1}$ which represents an over-determined system of linear equations (Pinar, 2001) from low dimension to high dimension. Second, for common wavelet kernels (including diffusion kernel here), their ranges are strictly included in $(-1,\ 1)$ that makes the eigenvalues of inverse GWT operator satisfy $\lambda_{\text{GWT}} > 1$ (Stéphane, 2009). Thus, the application of the inverse operator will substantially amplify the data in the GNN model, leading to a serious numerical instability error.

Model Complexity. As proposed in (Hammond et al., 2011), the GWT can be efficiently computed with polynomial approximation before training. Thus, the GNN model applying our framework keeps the same level of time complexity and the number of parameters as the original model, indicating the implementation-friendliness of our method.

5. Empirical Studies

We conduct experiments to answer the following research questions:

(**Q1**) How universal and effective is our enhancement framework in different GNNs as compared to the original models?

(**Q2**) Do the improvements of GNN models owe to the wavelet enhancement framework?

(**Q3**) What does the optimal scale(s) depends on? Is it necessary to apply multiple scales (compared to a single scale)?

5.1. Experiment Settings.

Datasets and Splits. We conduct experiments on widely-used real-world datasets. Specifically, we include Citation networks (Sen et al., 2008; Yang et al., 2016) : {Cora, CiteSeer, PubMed}, and the Amazon co-purchase networks (McAuley et al., 2015) : {Computers, Photo}. We summarize the statistics of these datasets in Appendix C.1.

For Citation networks, we use the standard splits in (Yang et al., 2016) and perform semi-supervised node classification tasks; for the Amazon co-purchase networks, we perform full-supervised node classification tasks, where we randomly divide the node set into train/validation/test sets with ratio 60%/20%/20%.

Baseline Models. We apply our enhancement framework on the following GNN models: GCN (Kipf & Welling, 2017), ChebNet (Defferrard et al., 2016), BernNet (He et al., 2021), ARMA (Bianchi et al., 2020), APPNP (Gasteiger et al., 2019), GAT (Veličković et al., 2018). Both spectral-based GNN and spatial-based GNN are included as baselines in extensive evaluations to verify the versatility of our method. All models are constructed with Pytorch Geometric (Fey & Lenssen, 2019) or publicly released code (BernNet).

Hyperparameter Settings. All GNN models are two-layers neural networks, which is a common setting in all baselines mentioned above. As our method is an enhancement framework, other hyperparameters will follow the original papers of baselines, except for ChebNet, ARMA, and BernNet (BernNet on Citation networks). For our framework, we use Chebyshev polynomials with 30 orders to approximate the graph wavelet operator. We set the number of scales to be 2 and apply grid search to obtain optimal scales. See Appendix E for details about hyperparameter settings.

Training and Testing. We use *Cross Entropy* as the loss function and Adam optimizer (Kingma & Ba, 2014) to train the models. For each dataset, we generate 10 random splits by random seeds, and then train and test all models on the same splits. We report the average metric for each model as the final test result. All experiments are trained and tested on a Linux server with an NVIDIA Tesla V100 GPU (32GB).

5.2. Experiments and Analysis

5.2.1. Node Classification (AQ1)

We evaluate our approach with a popular node classification task. Specifically, to intuitively show the effectiveness of our framework, we report the mean accuracy of both the wavelet enhancement version and the original version for each GNN model.

Analysis. The classification results are summarized in Table 3, where boldface letters are used to indicate the better one between the enhancement version and the original version. We can observe that our Multi-Resolution Wavelet Enhancement brings significant improvements to all GNN models, except for GAT on Cora and PubMed. Furthermore, we notice that the improvements achieved in learnable filters (ChebNet, ARMA, and BernNet) are generally greater than those achieved in pre-defined filters (GCN and APPNP). Besides, we find the random splits of Amazon co-purchase networks severely influence the testing results, which cause large variance and may hurt the reliability of the results in these two datasets (Computers and Photo). Despite all these influences, in general, our method is effective and universal for both spectral-based and spatial-based GNN models, which also gives an affirmative answer to the first research question.

To further verify the advantage of the proposed wavelet enhancement framework over GWNN, which implements GWConv with a mis-claim, we use ChebNet as the backbone and show the results on Citation networks in Table 2. We observe GWNN (reproduced with publicly released code) can only slightly outperform the original ChebNet, and even lead to worse performance in the PubMed network. In contrast, our method that implements correct GWConv significantly outperforms the original ChebNet in all these three datasets. The comparison shows the true power of GWConv in improving the expressiveness of GNN models.

Table 2. Comparison of GWNN, ChebNet, and ChebNet with wavelet enhancement. **Ours** denotes ChebNet with our enhancement framework.

	Cora	CiteSeer	PubMed
ChebNet	$80.50_{\pm 0.83}$	$67.40_{\pm 0.77}$	$77.40_{\pm 0.32}$
GWNN	$80.70_{\pm 0.92}$	$67.60_{\pm 0.92}$	$77.40_{\pm 0.62}$
Ours	$\mathbf{82.10_{\pm 0.72}}$	$\mathbf{68.50_{\pm 0.65}}$	$\mathbf{78.40_{\pm 0.39}}$

5.2.2. Ablation Studies (AQ2)

In this part, we conduct extensive ablation experiments to verify that the improvements owe a lot to the multi-scale graph wavelet rather than simply stacking multiple GNNs.

Table 3. Node classification results on real-world datasets. To each GNN model, we report the mean accuracy (%) of both wavelet enhancement (Model's name+Wavelet) and the original version.

Model	Cora	CiteSeer	PubMed	Computers	Photo
ChebNet	$80.50_{\pm 0.83}$	$67.40_{\pm 0.77}$	$77.40_{\pm 0.32}$	$87.08_{\pm 0.82}$	$\mathbf{92.61_{\pm 0.88}}$
ChebNet+Wavelet	$\mathbf{82.10_{\pm 0.72}}$	$\mathbf{68.50_{\pm 0.65}}$	$\mathbf{78.40_{\pm 0.39}}$	$\mathbf{87.62_{\pm 0.85}}$	$92.55_{\pm 1.07}$
ARMA	$80.70_{\pm 0.89}$	$67.70_{\pm 0.81}$	$77.30_{\pm 0.47}$	$87.13_{\pm 1.03}$	$87.38_{\pm 2.62}$
ARMA+Wavelet	$\mathbf{81.50_{\pm 0.72}}$	$\mathbf{68.60_{\pm 0.78}}$	$\mathbf{78.40_{\pm 0.45}}$	$\mathbf{87.94_{\pm 0.91}}$	$\mathbf{89.18_{\pm 2.21}}$
BernNet	$80.60_{\pm 1.01}$	$67.70_{\pm 0.96}$	$77.40_{\pm 0.44}$	$87.81_{\pm 0.95}$	$93.43_{\pm 1.35}$
BernNet+Wavelet	$\mathbf{81.40_{\pm 1.35}}$	$\mathbf{68.50_{\pm 1.08}}$	$\mathbf{78.10_{\pm 0.53}}$	$\mathbf{88.65_{\pm 1.13}}$	$\mathbf{93.79_{\pm 0.93}}$
APPNP	$80.80_{\pm 0.69}$	$68.10_{\pm 0.84}$	$77.70_{\pm 0.54}$	$85.16_{\pm 0.86}$	$88.95_{\pm 1.18}$
APPNP+Wavelet	$\mathbf{81.40_{\pm 0.46}}$	$\mathbf{68.50_{\pm 0.66}}$	$\mathbf{78.40_{\pm 0.33}}$	$\mathbf{86.63_{\pm 0.91}}$	$\mathbf{89.79_{\pm 0.93}}$
GCN	$80.60_{\pm 0.46}$	$67.90_{\pm 0.64}$	$77.90_{\pm 0.43}$	$\mathbf{87.82_{\pm 0.36}}$	$88.39_{\pm 0.70}$
GCN+Wavelet	$\mathbf{81.00_{\pm 0.38}}$	$\mathbf{68.30_{\pm 0.52}}$	$\mathbf{78.30_{\pm 0.31}}$	$87.43_{\pm 0.42}$	$\mathbf{88.41_{\pm 0.47}}$
GAT	$\mathbf{80.40_{\pm 0.72}}$	$67.30_{\pm 1.06}$	$\mathbf{76.70_{\pm 0.47}}$	$84.35_{\pm 2.61}$	$90.53_{\pm 1.56}$
GAT+Wavelet	$80.20_{\pm 0.44}$	$\mathbf{67.50_{\pm 0.82}}$	$76.60_{\pm 0.18}$	$\mathbf{85.01_{\pm 1.77}}$	$\mathbf{91.62_{\pm 1.63}}$

Specifically, we compare wavelet enhancement with 2 scales and a baseline that directly average two original GNNs. Due to space limitations, we only show results on ChebNet here and defer the full results to Appendix F.

Analysis. As shown in Table 4, compared to the original ChebNet, the direct average does not significantly boost its performance or even impair it. Similar results happen to other GNNs as shown in Appendix F, while these models achieve impressive performance with wavelet enhancement. Those results empirically prove that our framework does contribute to improvements in GNN models, which gives a positive answer to our second question at the same time.

Table 4. Ablation experiments on ChebNet. **Average** denotes the results of the direct average. **+Wavelet** denotes the results of wavelet enhancement.

	ChebNet	Average	+Wavelet
Cora	$80.50_{\pm 0.83}$	$80.30_{\pm 0.93}$	$\mathbf{82.10_{\pm 0.72}}$
CiteSeer	$67.40_{\pm 0.77}$	$67.00_{\pm 1.13}$	$\mathbf{68.50_{\pm 0.65}}$
PubMed	$77.40_{\pm 0.32}$	$77.40_{\pm 0.45}$	$\mathbf{78.40_{\pm 0.39}}$
Computers	$87.08_{\pm 0.82}$	$86.93_{\pm 0.81}$	$\mathbf{87.62_{\pm 0.85}}$
Photo	$\mathbf{92.61_{\pm 0.88}}$	$92.03_{\pm 0.82}$	$92.55_{\pm 1.07}$

5.2.3. Key Hyperparameter Studies(AQ3)

To answer the third question, in this section, we examine the influences caused by the scale s, the key hyperparameter of the framework. We conduct the experiments on three citation networks. In particular, we set the number of scales to be 2 and the range of each scale to be $[0.25, 4]$ with interval 0.25, leading to 256 scale pairs. For each dataset, we evaluate each GNN model on each scale pair, and then report the best 3 pairs of scales for each GNN model to avoid special cases.

Analysis. As shown in Figure 3, the optimal choices of scales are robust to different GNN models but largely depend on the datasets. Furthermore, we notice that the two scales are different in most optimal pairs (especially for the Cora network), which indicates the necessity of multiple scales.

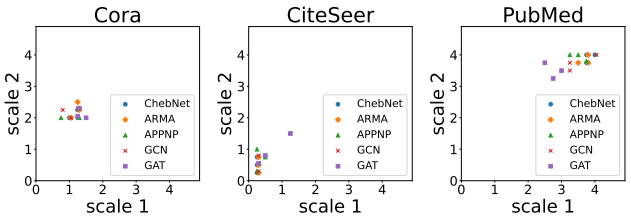

Figure 3. Results of hyperparameter studies. The figure indicates that the optimal scale strongly depends on the datasets and the necessity of multiple scales (MRA).

6. Conclusion

In this paper, we study a mis-claim about GWConv in (Xu et al., 2019) and derive the correct formulation for it. We further propose a GNN implementation to GWConv called Multi-Resolution Wavelet Enhancement, a universal framework that substantially improves GNN performance with marginal extra computational complexity. Due to the strong versatility and effectiveness of our method, we can easily achieve great improvements to GNN with a simple wavelet-based tool rather than involving more complex architecture. This may provide an alternative solution to GSP problems on large graphs.

However, the key hyperparameter, scale, is manually selected, which may lead to a bit of inconvenience. Moreover, the diffusion kernel applied in our framework is non-orthogonal which may make the filter term less powerful. For future work, a worth-doing direction is to investigate

the relationship between the dataset and GWT, and further develop an adaptive scales mechanism with a more effective kernel to improve the capability of the enhancement framework.

References

Allen, J. and Rabiner, L. A unified approach to short-time fourier analysis and synthesis. *Proceedings of the IEEE*, 65(11):1558–1564, 1977. doi: 10.1109/PROC.1977.10770.

Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. doi: 10.1126/science.286.5439.509. URL https://www.science.org/doi/abs/10.1126/science.286.5439.509.

Bathellier, B., Van De Ville, D., Blu, T., Unser, M., and Carleton, A. Wavelet-based multi-resolution statistics for optical imaging signals: Application to automated detection of odour activated glomeruli in the mouse olfactory bulb. *NeuroImage*, 34(3):1020–1035, 2007. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2006.10.038. URL https://www.sciencedirect.com/science/article/pii/S1053811906010743.

Bianchi, F. M., Grattarola, D., Livi, L., and Alippi, C. Graph neural networks with convolutional arma filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3496–3507, 2020. doi: 10.1109/TPAMI.2021.3054830.

Bruna, J., Zaremba, W., Szlam, A., and Lecun, Y. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014)*, 2014.

Chedemail, E., de Loynes, B., Navarro, F., and Olivier, B. Large graph signal denoising with application to differential privacy. *IEEE Transactions on Signal and Information Processing over Networks*, 8:788–798, 2022. doi: 10.1109/TSIPN.2022.3205555.

Cheung, G., Magli, E., Tanaka, Y., and Ng, M. K. Graph spectral image processing. *Proceedings of the IEEE*, 106(5):907–930, 2018. doi: 10.1109/JPROC.2018.2799702.

Chui, C. K. and ao Lian, J. A study of orthonormal multi-wavelets. *Applied Numerical Mathematics*, 20(3):273–298, 1996. ISSN 0168-9274. doi: https://doi.org/10.1016/0168-9274(95)00111-5. URL https://www.sciencedirect.com/science/article/pii/0168927495001115.

Chung, F. *Spectral Graph Theory*, volume 92. CBMS Regional Conference Series in Mathematics, 1997. ISBN 978-0-8218-0315-8. doi: /10.1090/cbms/092.

Cubuk, E. D., Zoph, B., Mané, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 113–123, 2019. doi: 10.1109/CVPR.2019.00020.

Dai, X., Diao, Y., Gu, Q., and Han, D. Wavelets with frame multiresolution analysis. *Journal of Fourier Analysis and Applications*, 9:39–48, 01 2003. doi: 10.1007/s00041-003-0001-5.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering, 2016. URL https://arxiv.org/abs/1606.09375.

Ding, K., Xu, Z., Tong, H., and Liu, H. Data augmentation for deep graph learning: A survey, 2022. URL https://arxiv.org/abs/2202.08235.

Donati, G., Petrone, A., and Rega, N. Multiresolution continuous wavelet transform for studying coupled solute–solvent vibrations via ab initio molecular dynamics. *Phys. Chem. Chem. Phys.*, 22:22645–22661, 2020. doi: 10.1039/D0CP02495C. URL http://dx.doi.org/10.1039/D0CP02495C.

Dong, X., Thanou, D., Toni, L., Bronstein, M., and Frossard, P. Graph signal processing for machine learning: A review and new perspectives. *IEEE Signal Processing Magazine*, 37(6):117–127, 2020. doi: 10.1109/MSP.2020.3014591.

dos Santos, I. S., Hall Barbosa, C. R., da Silva, M. T. F., and de Oliveira Filho, O. B. Multiresolution wavelet analysis for noise reduction in impulse puncture voltage measurements. *Measurement*, 153:107416, 2020. ISSN 0263-2241. doi: https://doi.org/10.1016/j.measurement.2019.107416. URL https://www.sciencedirect.com/science/article/pii/S0263224119312837.

Doucoure, B., Agbossou, K., and Cardenas, A. Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data. *Renewable Energy*, 92:202–211, 2016. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2016.02.003. URL https://www.sciencedirect.com/science/article/pii/S0960148116301045.

Drayer, E. and Routtenberg, T. Detection of false data injection attacks in power systems with graph fourier transform. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 890–894, 2018. doi: 10.1109/GlobalSIP.2018.8646454.

Duncan, A. J. Powers of the adjacency matrix and the walk matrix. *The Collection*, 9:4–11, 2004.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E. A survey of data augmentation approaches for nlp, 2021. URL https://arxiv.org/abs/2105.03075.

Fey, M. and Lenssen, J. E. Fast graph representation learning with pytorch geometric, 2019. URL https://arxiv.org/abs/1903.02428.

Flandrin, P. Some aspects of non-stationary signal processing with emphasis on time-frequency and time-scale methods. In Combes, J.-M., Grossmann, A., and Tchamitchian, P. (eds.), *Wavelets*, pp. 68–98, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg. ISBN 978-3-642-97177-8.

Fu, G., Hou, Y., Zhang, J., Ma, K., Kamhoua, B. F., and Cheng, J. Understanding graph neural networks from graph signal denoising perspectives, 2020. URL https://arxiv.org/abs/2006.04386.

Gadde, A., Anis, A., and Ortega, A. Active semi-supervised learning using sampling theory for graph signals. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 492–501, 2014.

Gama, F., Isufi, E., Leus, G., and Ribeiro, A. Graphs, convolutions, and neural networks: From graph filters to graph neural networks. *IEEE Signal Processing Magazine*, 37 (6):128–138, 2020a. doi: 10.1109/MSP.2020.3016143.

Gama, F., Isufi, E., Leus, G., and Ribeiro, A. Graphs, convolutions, and neural networks: From graph filters to graph neural networks. *IEEE Signal Processing Magazine*, 37 (6):128–138, 2020b. doi: 10.1109/MSP.2020.3016143.

Ganesan, D., Greenstein, B., Estrin, D., Heidemann, J., and Govindan, R. Multiresolution storage and search in sensor networks. *ACM Transactions on Storage*, 1:277–315, 08 2005. doi: http://doi.acm.org/10.1145/1084779.1084780.

Gao, F., Wolf, G., and Hirn, M. Geometric scattering for graph data analysis. In *International Conference on Machine Learning*, pp. 2122–2131. PMLR, 2019.

Gasteiger, J., Bojchevski, A., and Günnemann, S. Combining neural networks with personalized pagerank for classification on graphs. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1gL-2A9Ym.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry, 2017. URL https://arxiv.org/abs/1704.01212.

Graps, A. An introduction to wavelets. *IEEE Computational Science and Engineering*, 2(2):50–61, 1995. doi: 10.1109/99.388960.

Grimmett, G. *Random Graphs*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2 edition, 2018. doi: 10.1017/9781108528986.012.

Hamilton, W. L. *Graph Representation Learning*, volume 14. Morgan and Claypool, 2020. doi: 10.1007/978-3-031-01588-5. URL https://doi.org/10.1007/978-3-031-01588-5.

Hammond, D. K., Vandergheynst, P., and Gribonval, R. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2010.04.005. URL https://www.sciencedirect.com/science/article/pii/S1063520310000552.

Han, X., Jiang, Z., Liu, N., and Hu, X. G-mixup: Graph data augmentation for graph classification, 2022. URL https://arxiv.org/abs/2202.07179.

He, M., Wei, Z., Huang, Z., and Xu, H. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=WigDnV-_Gq.

Isufi, E., Loukas, A., Simonetto, A., and Leus, G. Autoregressive moving average graph filtering. *IEEE Transactions on Signal Processing*, 65(2):274–288, 2017. doi: 10.1109/TSP.2016.2614793.

Kenlay, H., Thanou, D., and Dong, X. Interpretable stability bounds for spectral graph filters, 2021. URL https://arxiv.org/abs/2102.09587.

Kim, W. H., Singh, V., Chung, M. K., Hinrichs, C., Pachauri, D., Okonkwo, O. C., and Johnson, S. C. Multiresolutional shape features via non-euclidean wavelets: Applications to statistical analysis of cortical thickness. *NeuroImage*, 93:107–123, 2014. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2014.02.028. URL https://www.sciencedirect.com/science/article/pii/S1053811914001396.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Kondor, R. and Lafferty, J. Diffusion kernels on graphs and other discrete structures. *Proceedings of the Nineteenth International Conference on Machine Learning*, 11, 04 2002.

Lang, M. Allpass filter design and applications. *IEEE Transactions on Signal Processing*, 46(9):2505–2514, 1998. doi: 10.1109/78.709538.

Ma, Y., Liu, X., Zhao, T., Liu, Y., Tang, J., and Shah, N. A unified view on graph neural networks as graph signal denoising, 2020. URL https://arxiv.org/abs/2010.01777.

Mahmoodabadi, S., Ahmadian, A., Abolhasani, M., Eslami, M., and Bidgoli, J. Ecg feature extraction based on multiresolution wavelet transform. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pp. 3902–3905, 2005. doi: 10.1109/IEMBS.2005.1615314.

Mallat, S. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989. doi: 10.1109/34.192463.

Mandic, D. P., Rehman, N. u., Wu, Z., and Huang, N. E. Empirical mode decomposition-based time-frequency analysis of multivariate signals: The power of adaptive data analysis. *IEEE Signal Processing Magazine*, 30(6):74–86, 2013. doi: 10.1109/MSP.2013.2267931.

McAuley, J., Targett, C., Shi, Q., and Hengel, A. v. d. Image-based recommendations on styles and substitutes, 2015. URL https://arxiv.org/abs/1506.04757.

Micheli, G., Rosenthal, J., and Vettori, P. Linear spanning sets for matrix spaces. *Linear Algebra and its Applications*, 483:309–322, 2015. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2015.06.008. URL https://www.sciencedirect.com/science/article/pii/S0024379515003614.

Mitra, S. and Hirano, K. Digital all-pass networks. *IEEE Transactions on Circuits and Systems*, 21(5):688–700, 1974. doi: 10.1109/TCS.1974.1083908.

Narang, S. K. and Ortega, A. Perfect reconstruction two-channel wavelet filter banks for graph structured data. *IEEE Transactions on Signal Processing*, 60(6):2786–2799, 2012. doi: 10.1109/TSP.2012.2188718.

Newman, M. *Networks: An Introduction*. Oxford University Press, 03 2010. ISBN 9780199206650. doi: 10.1093/acprof:oso/9780199206650.001.0001. URL https://doi.org/10.1093/acprof:oso/9780199206650.001.0001.

Newman, M. and Watts, D. Renormalization group analysis of the small-world network model. *Physics Letters A*, 263(4):341–346, 1999. ISSN 0375-9601. doi: https://doi.org/10.1016/S0375-9601(99)00757-4. URL https://www.sciencedirect.com/science/article/pii/S0375960199007574.

Ortega, A., Frossard, P., Kovačević, J., Moura, J. M. F., and Vandergheynst, P. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018. doi: 10.1109/JPROC.2018.2820126.

Patane', G. Fourier-based and rational graph filters for spectral processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022. doi: 10.1109/TPAMI.2022.3177075.

Pinar, M. Ç. *Overdetermined systems of linear equationsOverdetermined Systems of Linear Equations*. Springer US, Boston, MA, 2001. ISBN 978-0-306-48332-5. doi: 10.1007/0-306-48332-7_377. URL https://doi.org/10.1007/0-306-48332-7_377.

Proakis, J. Digital signal processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(4):392–394, 1975. doi: 10.1109/TASSP.1975.1162707.

Qian, S. and Chen, D. Joint time-frequency analysis. *IEEE Signal Processing Magazine*, 16(2):52–67, 1999. doi: 10.1109/79.752051.

R., L. A. The non-existence of a wavelet function admitting a wavelet transform convolution theorem of fourier type. *Technical Report*, 1994. URL https://cir.nii.ac.jp/crid/1573387449883144064.

Regalia, P., Mitra, S., and Vaidyanathan, P. The digital all-pass filter: a versatile signal processing building block. *Proceedings of the IEEE*, 76(1):19–37, 1988. doi: 10.1109/5.3286.

Rinaldo, R., Taubman, D., and Zakhor, A. Applications of multi-resolution analysis to images. In *Proceedings of the Seventh Workshop on Multidimensional Signal Processing*, pp. 3.10–3.10, 1991. doi: 10.1109/MDSP.1991.639334.

Rioul, O. and Vetterli, M. Wavelets and signal processing. *IEEE Signal Processing Magazine*, 8(4):14–38, 1991. doi: 10.1109/79.91217.

Rong, Y., Huang, W., Xu, T., and Huang, J. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkx1qkrKPr.

Sandryhaila, A. and Moura, J. M. F. Discrete signal processing on graphs: Graph fourier transform. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6167–6170, 2013a. doi: 10.1109/ICASSP.2013.6638850.

Sandryhaila, A. and Moura, J. M. F. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656, 2013b. doi: 10.1109/TSP.2013.2238935.

Sandryhaila, A. and Moura, J. M. F. Discrete signal processing on graphs: Graph filters. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6163–6166, 2013c. doi: 10.1109/ICASSP.2013.6638849.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective classification in network data. *AI Magazine*, 29(3):93, 09 2008. doi: 10.1609/aimag.v29i3.2157. URL https://ojs.aaai.org/index.php/aimagazine/article/view/2157.

Sharpnack, J., Rinaldo, A., and Singh, A. Detecting anomalous activity on networks with the graph fourier scan statistic. *IEEE Transactions on Signal Processing*, 64(2):364–379, 2016. doi: 10.1109/TSP.2015.2481866.

Shi, J. and Moura, J. M. Topics in graph signal processing: Convolution and modulation. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 457–461, 2019. doi: 10.1109/IEEECONF44664.2019.9049012.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013. doi: 10.1109/MSP.2012.2235192.

Strang, G. *Linear algebra and its applications.* Belmont, CA: Thomson, Brooks/Cole, 2006.

Stéphane, M. *A Wavelet Tour of Signal Processing (Third Edition)*. Academic Press, 2009. ISBN 978-0-12-374370-1. doi: https://doi.org/10.1016/B978-0-12-374370-1.00021-5. URL https://www.sciencedirect.com/science/article/pii/B9780123743701000215.

Tremblay, N. and Borgnat, P. Graph wavelets for multiscale community mining. *IEEE Transactions on Signal Processing*, 62(20):5227–5239, 2014. doi: 10.1109/TSP.2014.2345355.

Unser, M. and Blu, T. Wavelet theory demystified. *IEEE Transactions on Signal Processing*, 51(2):470–483, 2003. doi: 10.1109/TSP.2002.807000.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Verma, V., Qu, M., Kawaguchi, K., Lamb, A., Bengio, Y., Kannala, J., and Tang, J. Graphmix: Improved training of gnns for semi-supervised learning, 2019. URL https://arxiv.org/abs/1909.11715.

V.Henderson, H. and S.R.Searle. The vec-permutation matrix, the vec operator and kronecker products: a review. *Linear and Multilinear Algebra*, 9(4):271–288, 1981. doi: 10.1080/03081088108817379. URL https://doi.org/10.1080/03081088108817379.

Watts, D. J. and Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021. doi: 10.1109/TNNLS.2020.2978386.

Xu, B., Shen, H., Cao, Q., Qiu, Y., and Cheng, X. Graph wavelet neural network. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1ewdiR5tQ.

Yamada, K. Graph filter transfer via probability density ratio weighting, 2022. URL https://arxiv.org/abs/2210.14633.

Yang, Z., Cohen, W. W., and Salakhutdinov, R. Revisiting semi-supervised learning with graph embeddings, 2016. URL https://arxiv.org/abs/1603.08861.

Zhao, T., Liu, G., Günnemann, S., and Jiang, M. Graph data augmentation for graph machine learning: A survey, 2022. URL https://arxiv.org/abs/2202.08871.

Zhu, X. and Rabbat, M. Approximating signals supported on graphs. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3921–3924, 2012. doi: 10.1109/ICASSP.2012.6288775.

A. Proof of Theorems

A.1. Proof of Proposition 3.1

(Proposition 3.1) *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, and two graph signals $\{\vec{x}, \vec{y}\}$, their GWConv $\vec{x} \ *_{\mathcal{G}} \ \vec{y}$ with (invertible-) wavelet kernel function $g(\cdot)$ and scale s is defined as follows:*

$$\vec{x} \ *_{\mathcal{G}} \ \vec{y} = (\psi_g^s)^{-1}\theta(\vec{y})\psi_g^s\vec{x}\,, \tag{22}$$

where $\theta(\vec{y})$ is a symmetric matrix determined by $\vec{y}$ with an expression as Eq. (13)*.*

Proof. Considering that the GWT with an invertible kernel is an invertible transformation (Hammond et al., 2011), we have

$$\vec{x} \ *_{\mathcal{G}} \ \vec{y} = (\psi_g^s)^{-1}\psi_g^s(\vec{x} \ *_{\mathcal{G}} \ \vec{y})\,. \tag{23}$$

We further involve Eq. (8) into the equation above and obtain

$$\begin{aligned}
\vec{x} \ *_{\mathcal{G}} \ \vec{y} &= (\psi_g^s)^{-1}\psi_g^s(\vec{x} \ *_{\mathcal{G}} \ \vec{y}) \\
&= (\psi_g^s)^{-1}\mathbf{U}diag[g(s\vec{\lambda})](\mathbf{U}^T\vec{y} \odot \mathbf{U}^T\vec{x}) \\
&= (\psi_g^s)^{-1}\mathbf{U}diag(\mathbf{U}^T\vec{y})diag[g(s\vec{\lambda})]\mathbf{U}^T\vec{x} \\
&= (\psi_g^s)^{-1}[\mathbf{U}diag(\mathbf{U}^T\vec{y})\mathbf{U}^T]\mathbf{U}diag[g(s\vec{\lambda})]\mathbf{U}^T\vec{x} \\
&= (\psi_g^s)^{-1}\theta(\vec{y})\psi_g^s\vec{x}\,.
\end{aligned} \tag{24}$$

□

A.2. Proof of Lemma 3.3

(Lemma 3.3) *Let $\vec{y}$ to be a random vector in $\mathbb{R}^N$. The range of $\theta(\vec{y})$ is determined to be $LMS(\left\{\vec{u_i}\vec{u_i}^T : i = 1, 2, ..., N\right\})$. Similarly, when $g(s\lambda_i) \neq 0, i = 1, 2, ..., N$, the range of $diag(\psi_g^s\vec{y})$ is $LMS(\{diag(\vec{u_i}) : i = 1, 2, ..., N\})$.*

Proof. Both $\theta(\vec{y})$ and $diag(\psi_g^s\vec{y})$ can be further derived as:

$$\theta(\vec{y}) = \mathbf{U}diag(\mathbf{U}^T\vec{y})\mathbf{U}^T = \sum_{i=1}^{N}\langle\vec{u_i}, \vec{y}\rangle \cdot \vec{u_i}\vec{u_i}^T\,, \tag{25}$$

$$diag(\psi_g^s\vec{y}) = \sum_{i=1}^{N}[g(s\lambda_i)\,\langle\vec{u_i}, \vec{y}\rangle] \cdot diag(\vec{u_i})\,. \tag{26}$$

We only give the proof of the former one (i.e., proof about the range of $\theta(\vec{y})$) here because the proofs of them are highly similar. With a comparison between Eq. (25) and definition of LMS in Def 3.2, the problem is equivalent to prove

$$(\langle\vec{u_1}, \vec{y}\rangle\,, ..., \langle\vec{u_N}, \vec{y}\rangle)^T \in \mathbb{R}^N, \textit{ when } \vec{y} \in \mathbb{R}^N.$$

Notably, $\mathbf{U}$ is an orthogonal matrix that makes its row vectors, $\vec{u_i}$, become a set of orthogonal basis vectors for the $\mathbb{R}^N$. Thus, the $\vec{y}$ can be represented as

$$\vec{y} = (\vec{u_1}, ..., \vec{u_N})\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix} = \mathbf{U}\vec{a}, \quad \vec{a} \in \mathbb{R}^N\,. \tag{27}$$

On the other hand, we notice that

$$(\langle\vec{u_1}, \vec{y}\rangle\,, ..., \langle\vec{u_N}, \vec{y}\rangle)^T = \mathbf{U}^T\vec{y}\,. \tag{28}$$

Combining Eq. (27) with Eq. (28), we have:

$$(\langle\vec{u_1}, \vec{y}\rangle\,, ..., \langle\vec{u_N}, \vec{y}\rangle)^T = \mathbf{U}^T\mathbf{U}\vec{a} = \vec{a} \in \mathbb{R}^N\,. \tag{29}$$

Thus, we proved that range of $\theta(\vec{y})$ is determined to be $LMS(\left\{\vec{u_i}\vec{u_i}^T : i = 1, 2, ..., N\right\})$, where $\vec{y} \in \mathbb{R}^N$. The proof to $diag(\psi_g^s\vec{y})$ is similar. □

A.3. Proof of Lemma 3.4

(Lemma 3.4) *The filter term $\theta(\vec{y})$ is a diagonal matrix* ***iff*** $\vec{y} = t\sum_{i=1}^{N}\vec{u}_i$, *where* t *is an arbitrary real number. Specifically, the diagonal matrix $\theta(\vec{y})$ is $t \cdot \mathbf{I}$.*.

Proof. We prove sufficiency and necessity respectively.

Necessity: We just need to combine $\vec{y} = t\sum_{i=1}^{N}\vec{u}_i$ with Eq. (25) and easily obtain $\theta(\vec{y}) = t \cdot \mathbf{I}$. Obviously, $\theta(\vec{y})$ is a diagonal matrix here.

Sufficiency: Assuming there exists $\vec{\hat{y}} \in \mathbb{R}^N$ which makes $\theta(\vec{\hat{y}})$ a diagonal matrix. For convenience, we denote the diagonal elements of $\theta(\vec{\hat{y}})$ by $\{p_1, ..., p_N\}$. With the definition of $\theta(\vec{\hat{y}})$, we obtain:

$$\begin{bmatrix} p_1 & & \\ & \ddots & \\ & & p_N \end{bmatrix} \mathbf{U} = \mathbf{U} \begin{bmatrix} \left\langle \vec{u_1}, \vec{\hat{y}} \right\rangle & & \\ & \ddots & \\ & & \left\langle \vec{u_N}, \vec{\hat{y}} \right\rangle \end{bmatrix}$$

This equation indicates:

$$\begin{aligned} p_i &= \left\langle \vec{u_1}, \vec{\hat{y}} \right\rangle = ... = \left\langle \vec{u_N}, \vec{\hat{y}} \right\rangle, \quad \forall i \in \mathbb{Z},\ 1 \le i \le N\,, \\ \left\langle \vec{u_j}, \vec{\hat{y}} \right\rangle &= p_1 = p_2 = ... = p_N, \quad \forall j \in \mathbb{Z},\ 1 \le j \le N\,. \end{aligned} \tag{30}$$

Thus, we obtain a key property as follows

$$p_i = \left\langle \vec{u_j}, \vec{\hat{y}} \right\rangle, \quad \forall i, j \in \mathbb{Z},\ 1 \le i, j \le N\,. \tag{31}$$

Therefore, both $\theta(\vec{\hat{y}})$ and $diag(\mathbf{U}^T\vec{\hat{y}})$ can be represented as one thing: $\hat{t} \cdot \mathbf{I}$, which is a diagonal matrix. Involving $\hat{t} \cdot \mathbf{I}$ into $\theta(\vec{\hat{y}})$, we can easily obtain $\vec{\hat{y}} = \hat{t}\sum_{i=1}^{N}\vec{u}_i$. □

A.4. Proof of Proposition 3.5

(Proposition 3.5) *When ignoring the all-pass kernel mentioned in Lem. 3.4, the intersection of the range of two filters, $\theta(\vec{y})$ and $diag(\psi_g^s\vec{y})$, is empty.*

Proof. When ignoring special all-pass cases in Lem. 3.4, the $LMS(\left\{\vec{u_i}\vec{u_i}^T\right\})$, which denotes the range of $\theta(\vec{y})$, will never include a diagonal matrix; on the other hand, elements in $LMS(\{diag(\vec{u_i})\})$, the range of $diag(\psi_g^s\vec{y})$, are all diagonal matrices. Hence, the intersection of the range of two filter terms, $\theta(\vec{y})$ and $diag(\psi_g^s\vec{y})$, is empty. □

Moreover, the wavelet operator ψ_g^s (with diffusion kernel) here is a non-singular matrix that makes both ψ_g^s and $(\psi_g^s)^{-1}$ equivalent to linear transformation (Strang, 2006). Thus, the intersection of the range of $(\psi_g^s)^{-1}diag(\psi_g^s\vec{y})\psi_g^s$ and $(\psi_g^s)^{-1}\theta(\vec{y})\psi_g^s$ becomes empty, leading to great difference between Eq. (12) and Eq. (14).

B. Multi-Resolution Analysis with Wavelets

In this section, we introduce the details of wavelet-based MRA which has been mentioned in Sec. 4.2.

B.1. Resolution, MRA, and Wavelets in Computer Vision

First proposed in computer vision, the resolution is defined for measuring the local variations of the images (Rinaldo et al., 1991). Because the objects usually have very different sizes which makes it possible to pre-define an optimal resolution for analysis, the multi-resolution analysis (MRA) becomes necessary for a more effective image processing (Mallat, 1989). The key of MRA is to decompose the input images with a set of basis functions and obtain a scale-invariant interpretation of the images, while it can be properly handled by wavelets. Thus, the wavelet-based method had become a major of MRA and led to a fast development and generalization of definition and application.

B.2. Wavelet-based MRA for Signal Processing

As an effective tool for conventional signal processing and analysis, wavelet transform (WT) can be seen as an enhancement of windowed FT (also called Short-Time FT, STFT) mentioned in Sec. 2.3. A well-known Heisenberg inequality says that the product of time-resolution Δt and frequency-resolution Δf is lower bounded as follows:

$$\Delta t \cdot \Delta f \geq \frac{1}{4\pi} \tag{32}$$

The equation above means that one can only trade time-resolution for frequency-resolution. However, the resolution of STFT is determined and fixed once the window function is chosen, which makes it impossible to analyze the signals in both domains with high resolution. Compared to the fixed scale of window function in STFT, WT applies multiple scales mechanism and correspondingly variable window functions in signals decomposition, which leads to a multi-scale view of the input signals, i.e. MRA. Although the uncertainty principle (Eq. (32)) still exists, we can now analyze the signals with high resolution in both two domains **respectively**, as clearly shown in Figure 4.

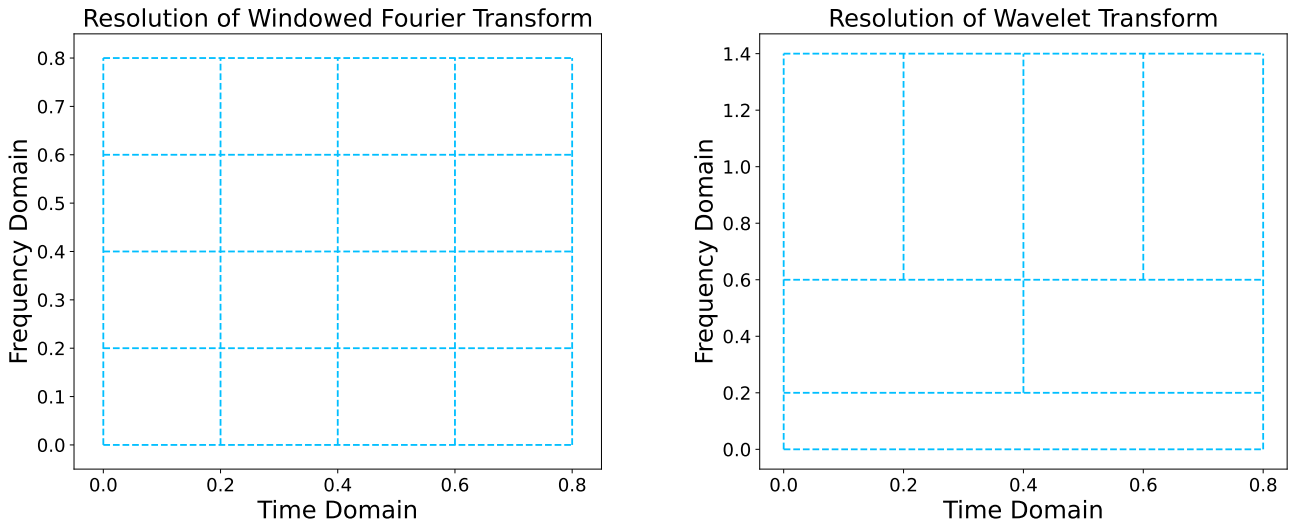

Figure 4. Different resolution of WT and STFT. Each rectangular area surrounded by dotted lines denotes a unit with the corresponding resolution, where it is fixed in STFT and variable in WT.

Such multiple resolutions effectively decompose the signals from both time aspect and scale (frequency) aspect which leads to more efficient time-scale (frequency) joint analysis (Qian & Chen, 1999; Mandic et al., 2013). All these have brought comprehensive analysis to the non-stationary and complex signal (Flandrin, 1989), and has achieved a lot of promising results in various real-life applications, such as wind-speed prediction (Doucoure et al., 2016), analysis of molecular mechanism (Donati et al., 2020), voltage measurements (dos Santos et al., 2020), neuro-image analysis (Bathellier et al., 2007; Kim et al., 2014), and so on.

C. Statistics of Datasets

In this section, we summarize the statistics of datasets used for computational experiments in Sec 3.3 and node classification tasks in Sec 5.

C.1. Real-World Datasets

The real-world datasets include three citation networks (Sen et al., 2008; Yang et al., 2016): {Cora, CiteSeer, PubMed}, and the Amazon co-purchase networks (McAuley et al., 2015): {Computers, Photo}.

The 20$\times$ for Citation networks here denotes 20 nodes per class. We split all node sets into train/validation/test with the rules mentioned in Table 5.

Table 5. Statistics of Real-World Datasets

	Cora	CiteSeer	PubMed	Computers	Photo
Nodes	2708	3327	19717	13752	7650
Edges	5278	4552	44324	245861	119081
Features	1433	3703	500	767	745
Classes	7	6	5	10	8
Node Splits	20×/500/1000	20×/500/1000	20×/500/1000	60%/20%/20%	60%/20%/20%

For Citation networks, we use split as (Yang et al., 2016) and perform semi-supervised node classification tasks; for the Amazon co-purchase networks, we perform full-supervised node classification tasks, where we randomly divide the node set into train/validation/test sets with ratio 60%/20%/20%.

C.2. Synthetic Datasets

The synthetic datasets come from popular network-generated models, including ER Random Graphs (Grimmett, 2018), BA Random Graphs (Barabási & Albert, 1999), Newman-Small-World Random Graphs (Newman & Watts, 1999) and original Small-World Random Graphs (Watts & Strogatz, 1998). We report the statistics of these datasets in Table 6.

Table 6. Statistics of Synthetic Datasets

	Newman-SW	Small-World	AB	ER
Nodes	100∼5000	100∼5000	100∼5000	100∼5000
Edges	**Random**	**Random**	**Random**	**Random**
Features	**Gaussian**	**Gaussian**	**Gaussian**	**Gaussian**

here, 100∼5000 of nodes denotes we generate random graphs from 100 nodes to 5000 nodes; **Random** of edges denotes the number of edges is random for each generated graph and depends on related factors, such as link probability, average degree, and so on; **Gaussian** of features denotes the signals on graphs are generated as random Gaussian variants.

D. Computational Experiments: Distance to "Ground-truth" Graph Convolution

In this section, we present the full results of computational experiments mentioned in Sec. 3.3. We conduct the computational experiments on both synthetic and real-world datasets which have been particularly introduced in Appendix C.

D.1. Results on Synthetic Datasets

For each synthetic dataset, we perform the graph convolution distance from two proposed GWConv to the "ground-truth" convolution (using GFT) with various scales and graph-related attributes, e.g. node's degree. Specifically, we generate synthetic graphs with a minimum of 100 and a maximum of 5000 nodes, and set an equispaced interval 200. We set the edge probability to be 0.5 for all synthetic datasets except for ER random graphs, while it is selected over $\{0.3, 0.7\}$ for ER graphs. We generate graph signal $\vec{x}$ and filter term $\vec{y}$ with **Gaussian** distribution as mentioned in Appendix C.2. We compute the graph convolution of two proposed GWConv with scale over $\{0.5, 1.5, 3.5\}$ and the attaching ratio over $\{0.3, 0.7\}$, and then compute graph convolution distance for each "scale-ratio" pair. All results are shown from Figure 5 to Figure 8.

D.2. Results on Real-World Datasets

We conduct experiments on real-world datasets which have been mentioned in Appendix C.1 with a scale over $\{0.5, 1.5, 3.5, 5\}$. Similar to the settings on synthetic datasets, we generate the filter term $\vec{y}$ with **Gaussian** distribution and then compute the Frobenius-norm of the difference of convolution results of proposed GWConv and convolution results applying GFT. The result is intuitively shown in Figure 9.

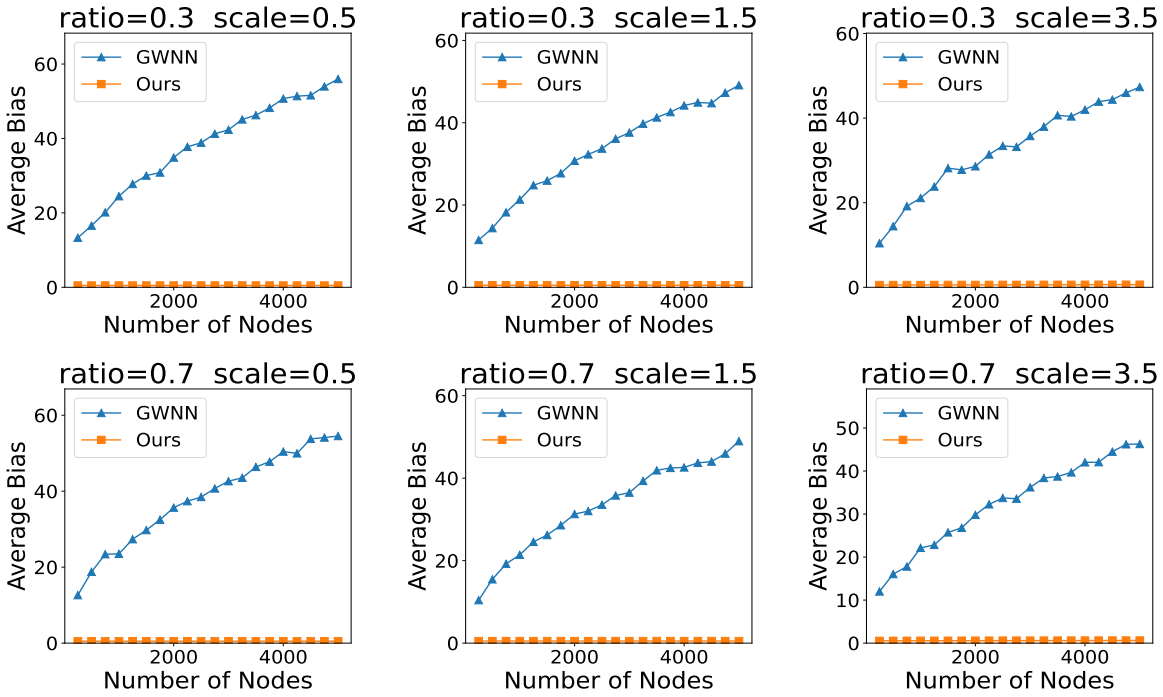

Figure 5. Results on *Erdős-Rényi* (ER) graphs

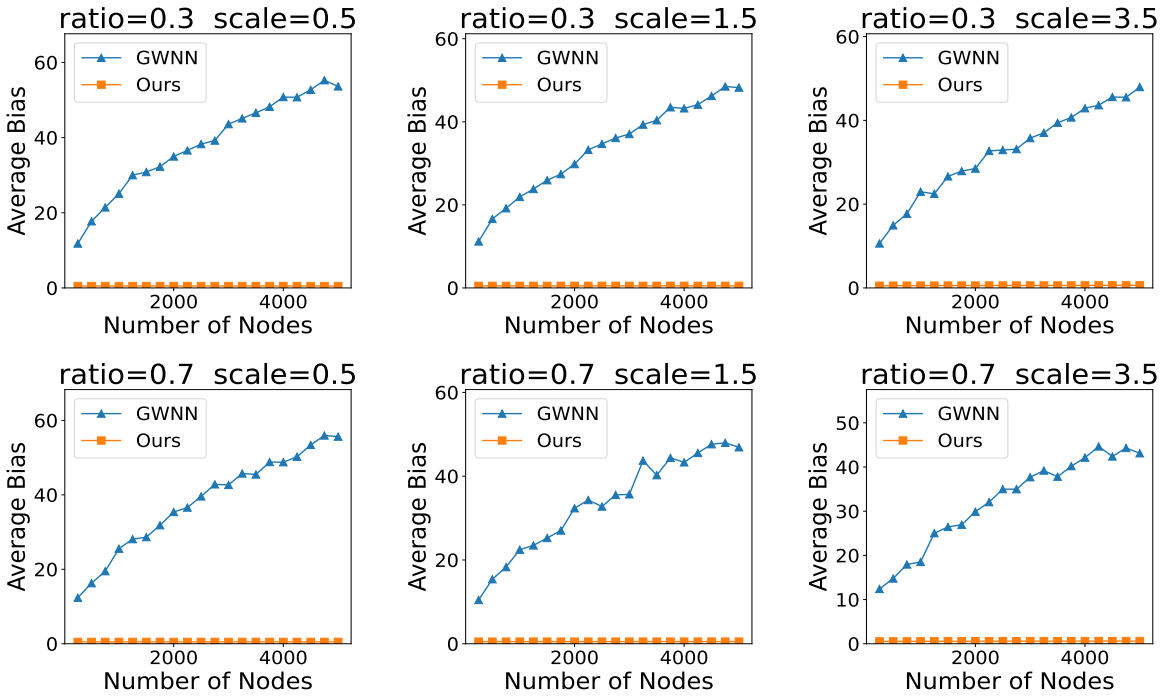

Figure 6. Results on *Barabási-Albert* (BA) graphs

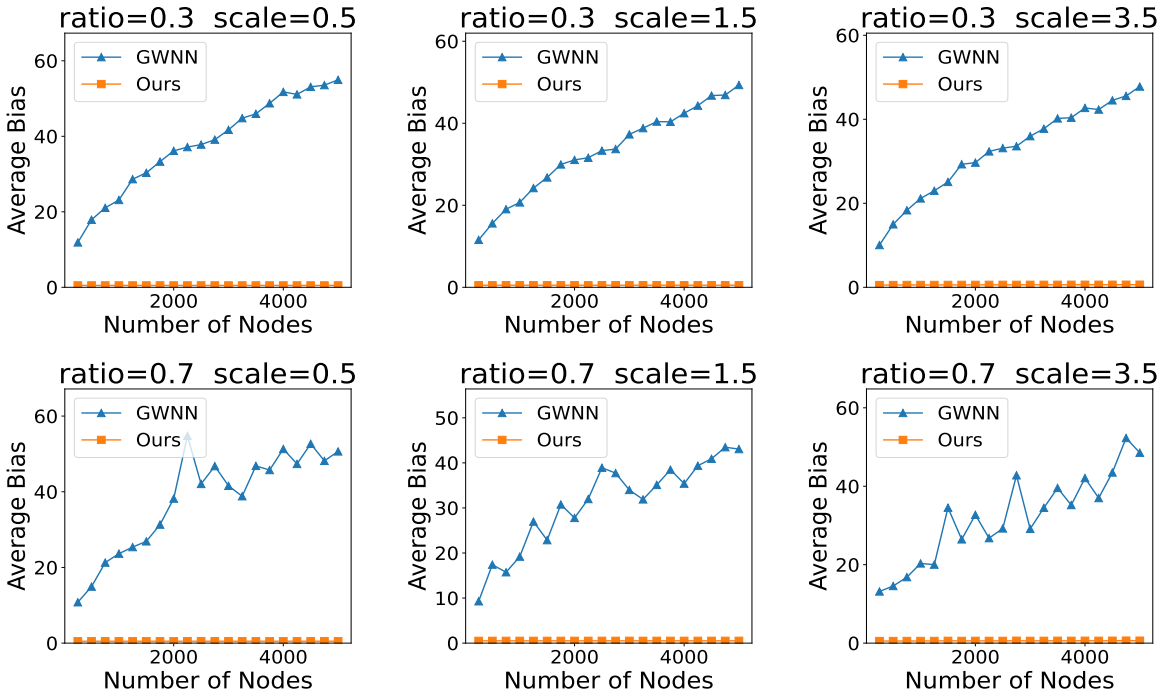

Figure 7. Results on $Newman\ Small\text{-}World$ graphs

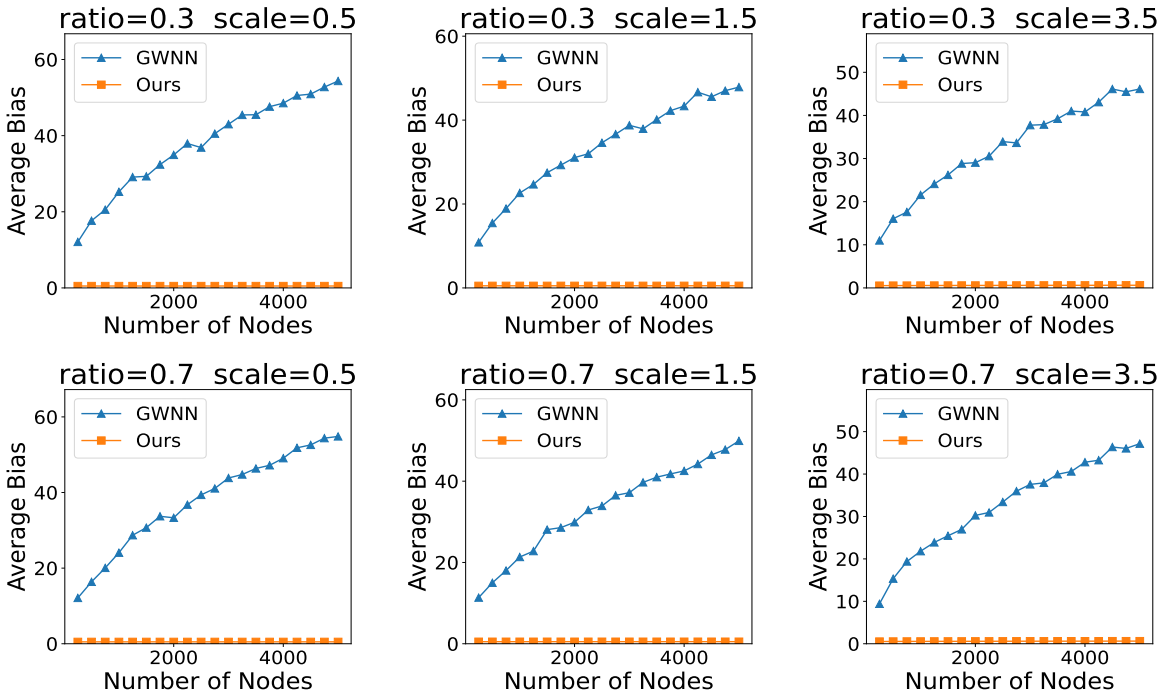

Figure 8. Results on original $Small\text{-}World$ graphs.

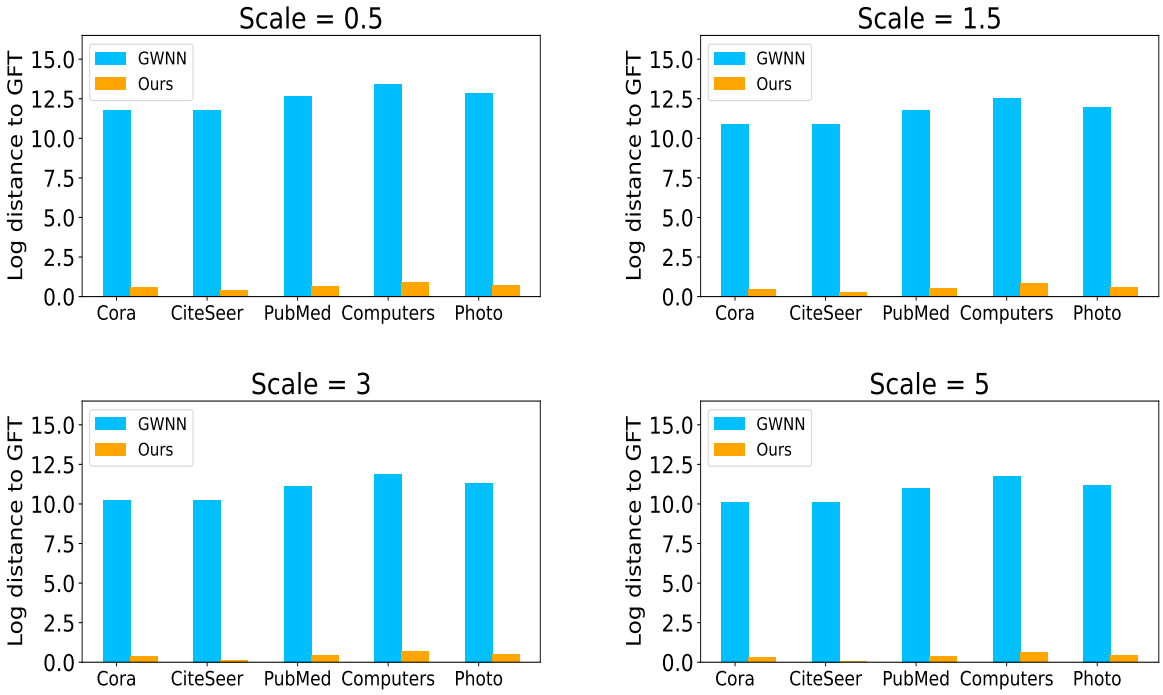

Figure 9. Results on Real-World Datasets

E. Hyperparameter Settings

We set a maximum of 2000 epochs and apply early stopping if validation accuracy doesn't increase for 200 epochs. All GNN models are 2-layers neural networks, and other hyperparameters (e.g., the K in APPNP) will follow the original papers of baselines, except for ChebNet, ARMA, and BernNet (BernNet on Citation networks). For ChebNet, we set approximation order K to be 2 for each layer and use 32 hidden units. For ARMA, we set both stack T and layer K to be 1 for each layer, and similarly use 32 hidden units. For BernNet, since the original BernNet paper only performs full-supervised node classification tasks, we follow the settings in the original paper for the experiments on Amazon networks, but set approximation order K to be 10 and use 32 hidden units for the experiments on Citation networks. For ChebNet, ARMA, and BernNet (BernNet on Citation networks), we set the dropout rate to be 0.5, the learning rate to be 0.01, and the weight decay to be 0.0005. All hyperparameter settings of baselines are intuitively shown in Table 7.

For our framework, we use Chebyshev polynomials with 30 orders to approximate the graph wavelet operator and set the number of scales to be 2. For each experiment, we first apply grid search over $\{0.25, 0.5, 0.75, ..., 3.5, 3.75, 4.0\}$ to obtain the optimal pair of scales with the highest accuracy on the validation set, and then evaluate our framework on the test set with such optimal scale pair.

Table 7. Hyperparameter Settings of GNN Models

GNN Model	Detailed Hyperparameter Settings
ChebNet	Hidden units= 32, $K = 2$, Dropout= 0.5, Learning rate= 0.01, Weight decay= 0.0005
ARMA	Hidden units= 32, $K = 1$, $T = 1$, Dropout= 0.5, Learning rate= 0.01, Weight decay= 0.0005
BernNet (Citation networks)	Hidden units= 32, $K = 10$, Dropout= 0.5, Learning rate= 0.01, Weight decay= 0.0005
BernNet (Amazon networks)	Following (He et al., 2021)
APPNP	Following (Gasteiger et al., 2019)
GCN	Following (Kipf & Welling, 2017)
GAT	Following (Veličković et al., 2018)

F. Ablation Experiments

We conduct the ablation experiments as a comparison of results between the wavelet enhancement (with 2 scales) and direct average of two original GNNs. We use boldface letters to indicate the better one. All results are shown in Table 8.

Table 8. Results of ablation studies.

Model	Cora	CiteSeer	PubMed	Computers	Photo
ChebNet					
Direct Average	$80.30_{\pm 0.93}$	$67.00_{\pm 1.13}$	$77.40_{\pm 0.45}$	$86.93_{\pm 0.81}$	$92.03_{\pm 0.82}$
+Wavelet	$\mathbf{82.10_{\pm 0.72}}$	$\mathbf{68.50_{\pm 0.65}}$	$\mathbf{78.40_{\pm 0.39}}$	$\mathbf{87.62_{\pm 0.85}}$	$\mathbf{92.55_{\pm 1.07}}$
ARMA					
Direct Average	$80.90_{\pm 0.49}$	$67.90_{\pm 0.86}$	$77.60_{\pm 0.48}$	$87.26_{\pm 0.85}$	$88.71_{\pm 3.01}$
+Wavelet	$\mathbf{81.50_{\pm 0.72}}$	$\mathbf{68.60_{\pm 0.78}}$	$\mathbf{78.40_{\pm 0.45}}$	$\mathbf{87.94_{\pm 0.91}}$	$\mathbf{89.18_{\pm 2.21}}$
BernNet					
Direct Average	$80.30_{\pm 1.32}$	$67.60_{\pm 1.56}$	$77.50_{\pm 0.43}$	$87.43_{\pm 1.19}$	$93.52_{\pm 0.98}$
+Wavelet	$\mathbf{81.40_{\pm 1.35}}$	$\mathbf{68.50_{\pm 1.08}}$	$\mathbf{78.10_{\pm 0.53}}$	$\mathbf{88.65_{\pm 1.13}}$	$\mathbf{93.79_{\pm 0.93}}$
APPNP					
Direct Average	$80.90_{\pm 0.39}$	$68.10_{\pm 0.66}$	$77.70_{\pm 0.32}$	$85.33_{\pm 1.22}$	$88.21_{\pm 0.94}$
+Wavelet	$\mathbf{81.40_{\pm 0.46}}$	$\mathbf{68.50_{\pm 0.66}}$	$\mathbf{78.40_{\pm 0.33}}$	$\mathbf{86.63_{\pm 0.91}}$	$\mathbf{89.79_{\pm 0.93}}$
GCN					
Direct Average	$80.40_{\pm 0.77}$	$67.50_{\pm 0.78}$	$77.50_{\pm 0.38}$	$86.77_{\pm 0.59}$	$88.21_{\pm 0.61}$
+Wavelet	$\mathbf{81.00_{\pm 0.38}}$	$\mathbf{68.30_{\pm 0.52}}$	$\mathbf{78.30_{\pm 0.31}}$	$\mathbf{87.43_{\pm 0.42}}$	$\mathbf{88.41_{\pm 0.47}}$
GAT					
Direct Average	$80.00_{\pm 1.03}$	$67.50_{\pm 0.88}$	$\mathbf{76.70_{\pm 0.36}}$	$83.91_{\pm 2.42}$	$90.61_{\pm 1.46}$
+Wavelet	$\mathbf{80.20_{\pm 0.44}}$	$\mathbf{67.50_{\pm 0.82}}$	$76.60_{\pm 0.18}$	$\mathbf{85.01_{\pm 1.77}}$	$\mathbf{91.62_{\pm 1.63}}$