WaveNet: Tackling Non-Stationary Graph Signals via Graph Spectral Wavelets

Zhirui Yang[1] , **Yulan Hu**[1,2] , **Sheng Ouyang**[1,2] , **Jingyu Liu**[1] , **Shuqiang Wang**[3] , **Xibo Ma**[4,†] ,
Wenhan Wang[5] , **Hanjing Su**[5] , **Yong Liu**[1,†]

[1]Renmin University of China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[4]Institute of Automation, Chinese Academy of Sciences (CASIA)
[5]Tencent Inc.
{yangzhirui,huyulan,ouyangsheng,liujy1016}@ruc.edu.cn, sq.wang@siat.ac.cn, xibo.ma@ia.ac.cn,
{wangwenhan88,hanjingsu}@gmail.com, liuyonggsai@ruc.edu.cn

Abstract

In the existing spectral GNNs, polynomial-based methods occupy the mainstream in designing a filter through the Laplacian matrix. However, polynomial combinations factored by the Laplacian matrix naturally have limitations in message passing (e.g., over-smoothing). Furthermore, most existing spectral GNNs are based on polynomial bases, which struggle to capture the high-frequency parts of the graph spectral signal. Additionally, we also find that even increasing the polynomial order does not change this situation, which means polynomial-based models have a natural deficiency when facing high-frequency signals. To tackle these problems, we propose *WaveNet*, which aims to effectively capture the high-frequency part of the graph spectral signal from the perspective of wavelet bases through reconstructing the message propagation matrix. We utilize Multi-Resolution Analysis (MRA) to model this question, and our proposed method can reconstruct arbitrary filters theoretically. We also conduct node classification experiments on real-world graph benchmarks and achieve superior performance on most datasets. Our code is available at https://github.com/Bufordyang/WaveNet

1 Introduction

Graph, as an abstract data structure, can efficiently represent complex relational structures between information. Therefore, it has been widely used in real-life scenarios. With the advent of the deep learning era in artificial intelligence, graph neural networks (GNNs) are specifically designed to address graph-related problems such as traffic network prediction (Bogaerts et al. 2020; Cui et al. 2020a), social network recommendation (Do et al. 2022; Gao et al. 2022), molecular structure analysis (Jiang et al. 2021; Yang et al. 2019), and more (Kipf and Welling 2017; Hamilton, Ying, and Leskovec 2017; Veličković et al. 2018; Xu et al. 2019b).

Since a graph is composed of vertices set and edges set, existing GNN methods can be roughly divided into spatial domain GNNs and spectral domain GNNs. The former mainly learn node representations through message passing paradigm or attention network structure (Wu et al. 2021; Zhang et al. 2020; Rong et al. 2020). The latter utilizes the laplacian matrix of the graph for spectral analysis, designing networks by mapping the feature matrix to the spectral domain through graph Fourier transform. While spatial GNNs have demonstrated promising results on homogeneous graphs, they're still tricky to deal with heterogeneous graphs. This is because the message passing paradigm is based on the homogeneity hypothesis, which means the same class nodes mostly links with same class neighbors. However, in the case of a heterogeneous graph, aggregating the information from neighbor nodes will destroy the model's expressive ability (Zhu et al. 2021). Spectral GNNs focus on dealing heterogeneous graph via designing a filter in the spectral domain of the graph Laplacian matrix (Kipf and Welling 2017; Defferrard, Bresson, and Vandergheynst 2016; Chien et al. 2021). The effectiveness of these filters directly impacts the performance of downstream tasks. And most methods rely on truncated polynomial bases encounter difficulties in fitting high-frequency spectral signals (Feng et al. 2022). We aim to improve this situation by using non-polynomial bases.

Wavelets, as a powerful mathematical tool for analyzing time-varying non-stationary signals, have received extensive attention in signal processing and feature extraction (Guo et al. 2022). Wavelet bases are mostly composed of signals with finite energy, which gives them a natural advantage over polynomial bases when the signal changes drastically (Boggess and Narcowich 2015). However, existing wavelet-based GNNs model usually utilize Chebyshev polynomials to approximate wavelets transform matrix (Hammond, Vandergheynst, and Gribonval 2011; Xu et al. 2019a; Donnat et al. 2018). However, this approach constrains the wavelet's ability to fit high-frequency graph spectral signals, resulting in previous work not achieving significant improvements. In wavelet analysis, Multi-Resolution Analysis (MRA) establishes a connection between time and frequency domains information. Therefore, we perform MRA on graph spectral signals, and employ non-polynomial bases (wavelets) to capture finer changes of spectral signal.

On the other hand, we have found through experiments that in polynomial-based methods, the main contributing bases are often limited to only the first few terms (usu-

[†]Corresponding author

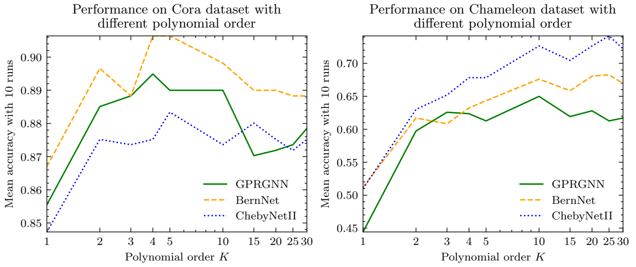

Figure 1: Different polynomial order K applied to the Cora and Chameleon graph dataset. Mean accuracy is under 10 runs and the variance term is omitted.

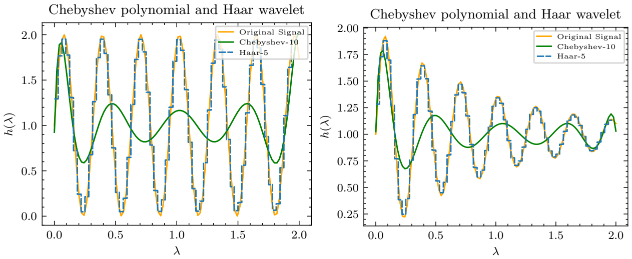

Figure 2: The illustration about the ability of 10-order Chebyshev polynomial and 5-scale Haar wavelet to fit high-frequency signals, which are $\sin 20\lambda + 1$ and $\sin 20\lambda \cdot e^{-\lambda} + 1$ respectively.

ally orders 5 to 10 as Figure 1 showing), and the performance of these methods even appear to decrease with increasing polynomial order, which is counter-intuitive and violates the theory of polynomial approximation (Powell 1981). We have a deep thought about why this phenomenon appear: 1) Assuming the optimal filter's shape is complex and unsmooth, then increasing the polynomial order should improve the situation according to polynomial approximation theory. However, Figure 1 demonstrates that this approach loses efficiency. We believe that this happens because polynomial-based models are insufficient to learn complex filter. We summarize this phenomenon as insufficient capacity of the model; 2) On the opposite, the optimal filter's shape is simple and smooth, which makes polynomial-based models reach a bottleneck. About the truth of whether the optimal filter is smooth or unsmooth still wait to solve, and previous researches have shown that most heterogeneous graph datasets have a unsmooth filter (He et al. 2021; He, Wei, and Wen 2022) in experiments, such as band-rejection and comb filters. In Specformer (Bo et al. 2023), the results show that learning more complicated filters, which exhibit dramatic changes, is helpful and achieves better performance. As a result, we suggest that polynomial-based models tend to learn smooth filters and struggle to cope with spectral changes with high-variance. For concisely, we call it as high-frequency spectral signals filtering. And we explore the wavelet bases for filtering to tackle this problem.

Before that, we will introduce a concept originating from the Computer Vision (CV) domain: *scale*, which is differentiate from its traditional meaning in the graph domain. In the CV field, *scale* typically refers to the appearance of images or objects at different observation distances and resolutions (Lowe 2004), which is distinct from the concept of scalability in graphs. In this paper, *scale* refers to the components of graph spectral signals at different resolutions as the same conception in CV. Finer scale represents the higher frequency components of graph spectral signal.

Therefore, we call *polynomial bases are scaleless*, like but not Gibbs-Wilbraham phenomenon (Hewitt and Hewitt 1979) (overshoot occurs in discontinuous points). It is more like underfitting. To validate our view, we conduct an additional synthetic graph experiment to explore the scale of polynomials. We created a synthetic toy graph consisting of 50 nodes, and node features are randomly sampled from Gaussian distribution. We apply a pre-determined high-frequency spectral signal f to generate the nodes labels. Our goal is to reconstruct the pre-determined signal f. In Figure 2, yellow curve is the pre-determined spectral signal. We use the 10-order chebyshev polynomials and the 5-scale Haar wavelet to reconstruct the original signal. As the Figure 2 showing: low-order polynomials bases are extremely struggle to fit the high frequency signal. While the Haar wavelet (Haar 1909) fits well. The result demonstrates that wavelets are more effective in handling high-frequency signal. While polynomials' curve shows scaleless and only coarse part is represented. In a word, wavelets are born to solve the high frequency components of signals.

To address those issue mentioned above, and capture scale information more effectively, we propose a novel graph neural network called *WaveNet*. And our contributions are summarized below:

- **We perform MRA on graph spectral signals:** As a result, we use the scaling function of Haar wavelet to reconstruct the graph spectral filter. Theoretically, wavelets have the capability to learn arbitrary graph spectral filters. And *WaveNet* excels at capturing high-frequency signals through the experiments results.

- **A novel model proposed *WaveNet*:** Our model leverages wavelet bases filters on graph spectral signals. Through our experiments, our model demonstrates the capability to learn arbitrary graph spectral filters. Moreover, our experiments reveal *WaveNet* possess a stronger ability to capture scale information compared to polynomials-based models.

- **A new perspective of graph spectral filter:** We open a new discussion about non-polynomial and polynomial-based methods, and trying to explore non-polynomial approaches. We analyze the graph spectral signal through wavelets, and our experiments demonstrate the effectiveness and superior performance of the wavelet-based approach. This can provide researchers with further insight into graph spectral.

2 Preliminaries

2.1 Notations

We only consider in the undirected graph $\mathcal{G} = (\mathbb{V}, \mathbb{E}, \mathbf{X})$ with node set $\mathbb{V}$ and edge set $\mathbb{E}$. In the spectral GNNs, we usually treat node feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ as graph signal, where $\boldsymbol{x}(i)$ denotes the signal of node i. The graph structure information can be denoted as adjacency matrix $\mathbf{A}$, and $\mathbf{D}$ denote the diagonal degree matrix. Usually, we use $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ to denote the normalized adjacency matrix. Especially, the normalized Laplacian matrix can be denoted as $\mathbf{L} = \mathbf{I} - \tilde{\mathbf{A}}$, and the eigenvalues λ_i of $\mathbf{L}$ is in the interval $\lambda_i \in [0, 2]$. Let $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ denote the eigendecomposition of normalized Laplacian matrix, where $\mathbf{U}$ is the eigenvector matrix, $\mathbf{\Lambda} = diag[\lambda_1, \ldots, \lambda_n]$ is the diagonal eigenvalues matrix.

2.2 Graph Signal Filter and Spectral GNNs

For a graph signal $\mathbf{X}$, the graph Fourier transform is defined as $\hat{\mathbf{X}} = \mathbf{U}^\top\mathbf{X}$, and the inverse transform is $\mathbf{X} = \mathbf{U}\hat{\mathbf{X}}$ (Shuman et al. 2013). When filtering on spectral graph signals, we can utilize a filter function $g_\theta(\cdot)$ on the Laplacian matrix. This can be described as:

$$g_\theta(\mathbf{L})\mathbf{X} = \mathbf{U}g_\theta(\mathbf{\Lambda})\mathbf{U}^\top\mathbf{X}. \tag{1}$$

In earlier works, filter function $g_\theta(\cdot)$ is represented by using various polynomial basis:

$$g_\theta(\lambda_i) = \sum_{k=0}^{K-1} c_k \mathrm{Poly}_k(\lambda_i), \tag{2}$$

$$g_\theta(\mathbf{L})\mathbf{X} = \sum_{k=0}^{K-1} c_k \mathbf{U}\mathrm{Poly}_k(\mathbf{\Lambda})\mathbf{U}^\top\mathbf{X} = \sum_{k=0}^{K-1} c_k \mathrm{Poly}_k(\mathbf{L})\mathbf{X}.$$

2.3 Wavelet on Graph Neural Networks

Since the signals of graph are storing in the vertex domain, and mapping them to the time and frequency domains is the key to analyzing the graph signals. The spectral domain methods based on wavelet transform are proposed in (Hammond, Vandergheynst, and Gribonval 2011; Xu et al. 2019a; Cui et al. 2020b). The main contribution here lies in defining the graph wavelet matrix $\mathbf{\Psi}_s = \mathbf{U}\mathbf{G}_s\mathbf{U}^\top, \mathbf{G}_s = \mathrm{diag}(g(s\lambda_0), \ldots, g(s\lambda_{N-1})))$. And the graph convolution based on graph wavelet transform is defined as

$$\boldsymbol{x} \otimes_{\mathcal{G}} \boldsymbol{y} = \mathbf{\Psi}_s((\mathbf{\Psi}_s^{-1}\boldsymbol{x}) \odot (\mathbf{\Psi}_s^{-1}\boldsymbol{y})). \tag{3}$$

However, the graph wavelet matrix $\mathbf{\Psi}_s$ relies on Chebyshev polynomial approximation (Xu et al. 2019a). Therefore this approach is still under polynomial bases which limits the representation ability of wavelet. Different wavelets have different expression function, and most wavelet corresponds a "mother" wavelet function $\psi(x)$ and a scaling function $\phi(x)$. The $\phi(sx - t)$ means that scaling function at scale s and translation at t. For the same scale mother function $\psi(sx)$ and scaling function $\phi(sx)$ satisfy $\int_t \psi(st)\phi(st)dt = 0$, which means they are orthonormal in $L^2(\mathbb{R})$ space. And in next section, we will elaborate on how to perform MRA on graphs from the perspective of spatial approximation theory.

3 WaveNet

Wavelets are designed to represent finite-energy function in terms of specific building blocks at different scales and positions. The advantage of wavelets is to capture information about signals that change dramatically, which has significantly better performance than polynomial bases. In previous work (Hammond, Vandergheynst, and Gribonval 2011; Xu et al. 2019a), the wavelet transform on graph has been defined. However, the inner connection between spectral domain GNNs and wavelets has not been thoroughly explored. Specifically, there are still unexplored properties about the relationship between wavelets scales and graph filters. In this section, we use the MRA on graph to reconstruct the spectral filter.

3.1 Multi-Resolution Analysis of Wavelet On Graph

We define a finite-energy signal f which satisfies $||f||^2 \equiv \int_{-\infty}^{\infty} f^2(t)dt < \infty$, let signal $f(\cdot) \in L^2(\mathbb{R})$, and then we will introduce how to represent signal f in L^2 space and how to split L^2 space via MRA.

Assuming $\{V_j\}_{j\in\mathbb{Z}}$ is a subspace of $L^2(\mathbb{R})$ space, and we consider $\{V_j\}_{j\in\mathbb{Z}}$ is a MRA of $L^2(\mathbb{R})$ space. For each subspace, is strictly contained layer by layer with j increasing $V_{j-1} \subset V_j$, and the information including in subspace V_{j-1} is completely contained in V_j. The main idea of wavelet is dividing the $L^2(\mathbb{R})$ space into finite number of subspaces, and utilizing different scale wavelet basis to fit the information which contain in the subspace.

We denote the wavelet scaling function $\phi_{j,k}(\cdot) \in L^2(\mathbb{R})$ with the dilation j and translation k. The linear space V_j can be spanned by the set of integral translation of wavelet scaling function. As for the dyadic wavelets, it can be described as following:

$$V_j \equiv \{\ldots, \phi(2^j t + 1), \phi(2^j t), \phi(2^j t - 1), \ldots\}. \tag{4}$$

Notice that V_j is the space of piecewise constant functions of finite support whose discontinuities are contained in the set $\{\ldots - 1/2^j, 0, 1/2^j, 2/2^j, 3/2^j \ldots\}$, therefore we need another space to fill the deficiency of V_j in L^2 space.

Theorem 1 ((Boggess and Narcowich 2015)) *Let W_j be the space of all function form $\sum_{k\in\mathbb{Z}} b_k\psi(2^j t - k), b_k \in \mathbb{R}$, where $\psi_{j,k}(\cdot) \in L^2(\mathbb{R})$ is called "mother" wavelet and only finite number of* b_k *are nonzero, W_j is the orthogonal complement of V_j in V_{j+1} and we have:*

$$V_{j+1} = V_j \oplus W_j. \tag{5}$$

The space $L^2(\mathbb{R})$ can be decomposed as an infinite orthogonal direct sum

$$L^2(\mathbb{R}) = V_0 \oplus W_0 \oplus W_1 \oplus \ldots. \tag{6}$$

Therefore $f \in L^2(\mathbb{R})$ can be written uniquely as

$$\begin{aligned} f &= f_0 + \sum_{j=0}^{\infty} w_j \\ &= \sum_k a_k\phi(s_0x - k) + \sum_{j=0}^{\infty}\sum_k b_{jk}\psi(s_jx - k), \end{aligned} \tag{7}$$

$$a_k = \langle f(x), \phi(s_0x - k)\rangle, b_{jk} = \langle f(x), \psi(s_jx - k)\rangle, \tag{8}$$

where f_0 belongs to V_0 and w_j belongs to W_j, $a_k, b_{jk} \in \mathbb{R}$, and subspace V is obtained by the linear combination of scaling functions $\phi(\cdot)$, orthogonal complement space W is obtained by the linear combination of "mother" wavelet functions $\psi(\cdot)$.

However, the coefficients a_k, b_{jk} are difficult to obtain since the filter f is agnostic. Our goal is to learn a filter through neural network, thereby if we can learn the coefficients and then we can attain a specific filter. Intuitively to explain the Equation (7): if we make a MRA at resolution j, and employ MALLAT algorithm (Mallat 1989) with two channel tower decomposition on signal f, then f can be decomposed into detail component f_d (obtained by high-pass filter) and coarse component f_c (obtained by low-pass filter). By repeating this process on the coarse component, we have:

$$f = f_{d_0} + f_{d_1} + \ldots + f_{d_{j-1}} + f_{d_j} + f_{c_j}, \tag{9}$$

which decomposes f as a series combination of signals at different resolutions. The space V_j contains all the related information of signal at resolution j. If we choose a wavelet with finite or compact support (e.g.Haar wavelet), with the resolution increasing, the building block of haar basis will be smaller, and more detail information will be contained in the higher subspace. When $j \to +\infty$, each building block can be represented as an impulse signal $\delta(\cdot)$ in the V_∞ subspace. That means any signal can be fully represented under infinite resolution.

Proposition 1 *According to Theorem 1, the containment $V_{j-1} \subset V_j \subset V_{j+1}$ is strict, the lower subspace scale function $\phi(\cdot)$ and the "mother" wavelet $\psi(\cdot)$ can be composed by higher space scale function $\phi(\cdot)$ as follows:*

$$\phi(s_j x) = \sum\nolimits_k c_k \phi(s_{j+1}x - k), \tag{10}$$

$$\psi(s_j x) = \sum\nolimits_k c_k \phi(s_{j+1}x - k). \tag{11}$$

The orthogonal complement space W_j can use scaling functions instead of "mother" wavelet by using Equation (11). *Combining Equation* (7) *and Equation* (11), *the signal $f \in L^2(\mathbb{R})$ can be represented by the scaling function family:*

$$f(x) = \sum_k a_k \phi(s_0 x - k) + \sum_{j=0}^{\infty} \sum_k b_{jk} \phi(s_{j+1}x - k). \tag{12}$$

As a result, we propose to only use the scaling function to obtain a agnostic filter f. And Pro. 1 also embodies the idea of filtering. We can truncate Equation (12) at resolution s_k to filter out noise from $s_{j>k}$. To simplify the reconstruction process of the filter f, we chose the Haar wavelet as the basis with dyadic scale and shows in the next.

3.2 Spectral Filter Reconstruct via Scaling Functions

In this section, we will use the simplest Haar wavelet to reconstruct a finite-energy spectral filter $f(\lambda)$. The scaling function of Haar wavelet is defined as follows:

$$\phi(2^j\lambda - k) = \begin{cases} 1 & 0 \le 2^j\lambda - k < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

where, 2^j is called scaling or dilation parameter, k is the translation parameter. We use $\phi_{j,k}(\cdot)$ to denote dilation at scale 2^j and translation to position k, and it can yield an orthogonal basis in $L^2(\mathbb{R})$ as follows:

$$V_j \equiv \overline{\text{span}\{\phi_{j,k}(\cdot), k \in \mathbb{Z}\}}. \tag{14}$$

And $\psi_{j-1,k}(\cdot) \in V_{j-1}$ can be represented by $\phi_{j,k}(\cdot)$ as Equation (11). We aim to use the wavelet basis to reconstruct the filter function through scaling function of Haar wavelet. We denote the optimal filter is $f^*(\lambda)$, it can be written similarly to Equation (12) as follows:

$$f^*(\lambda) = \sum_k a_k \phi_{0,k}(\lambda) + \sum_{j=0}^{\infty} \sum_k b_{jk} \phi_{j+1,k}(\lambda). \tag{15}$$

Theoretically, we can use Equation (15) to reconstruct any functions. But in practice, it is not necessary to decompose a finite-energy signal at infinite resolution since finite-energy signal's highest component can be fully represented in the V_J subspace. This shows that we can truncate Equation (15) at resolution J. Combining with Equation (10) and Equation (15). Consequently, we have:

$$\begin{aligned} f^*(\lambda) &\approx \sum_{j=0}^{J} \sum_k \langle f^*(\lambda), \phi_{j,k}(\lambda) \rangle \phi_{j,k}(\lambda) \\ &= \sum_{j=0}^{J} \sum_k c_{jk} \phi_{j,k}(\lambda) \overset{(10)}{\underset{j \to J}{=\!=\!=}} \sum_k \theta_k \phi_{J,k}(\lambda), \end{aligned} \tag{16}$$

where θ_k is the coefficient leaerned by network. Let $\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ denote the eigendecomposition of normalized Laplacian matrix, where $\mathbf{U}$ is the eigenvector matrix, $\boldsymbol{\Lambda} = diag[\lambda_1, \ldots, \lambda_n]$ is the diagonal eigenvalues matrix. We use

$$\mathbf{PX} = \mathbf{U} \sum_{k=-\infty}^{+\infty} \theta_k \phi_{J,k}(\boldsymbol{\Lambda}) \mathbf{U}^\top \mathbf{X} \tag{17}$$

to reconstruct propagation matrix $\mathbf{P}$ and filter on graph signal $\mathbf{X}$, we aim for the the network can learn the correct coefficients θ_k through gradient descent. The *WaveNet* architecture detail is described in Appendix A.3.

3.3 Complexity

Our model has two steps of filtering: spectral decomposition and filter learning process. Since the complexity of the spectral decomposition is $\mathcal{O}\left(n^3\right)$, we perform the decomposition as a preprocessing step. The filter learning process can be separated as two part: reconstruction of propagation matrix and forward linear layers. The complexity of learning a propagation matrix by scaling function is $\mathcal{O}\left(n^2\right)$. And the forward linear layers' complexity is $\mathcal{O}\left(d \times h + h \times c\right)$, where d is the node feature dimensions, h corresponds the hidden dimensions, and c denotes the number of classes.

4 Related Work

4.1 Spatial GNNs

Spatial GNNs are based on message passing paradigm and aggregate graph information in the spatial domain. They usually focus on conducting various aggregation methods on vertex or edge domain to capture the graph information. GCN (Kipf and Welling 2017) is one of the earliest work on spatial domain and uses the graph adjacent matrix to aggregate one-hop neighbors. Geom-GCN (Pei et al. 2020) conducts geometric aggregation scheme to attain long-range dependencies in heterogeneous graphs. H2GCN (Zhu et al. 2020) model focus on improve the efficiency of neighbor aggregation. GGCN's (Yan et al. 2022) main idea is to make full use of the edges. This is achieved by assigning positive or negative signs to the edges, and makes a superior performance on heterogeneous graphs.

4.2 Spectral GNNs

Spectral GNNs are based on filtering graph signal in spectral domain. ChebyNet (Defferrard, Bresson, and Vandergheynst 2016) is based on Chebyshev polynomial to approximate a filter. APPAP (Klicpera, Bojchevski, and Günnemann 2019) conduct Personalized PageRank to attain the propagation matrix, and its a kind of fixed filter. GPR-GNN (Chien et al. 2021) is a learnable filter model by learning the Monomial basis combination coefficients. BernNet (He et al. 2021) utilizes the Bernstein basis to learn a filter function, which is more strong than monomial basis. JacobiConv (Wang and Zhang 2022) employs Jacobi polynomials basis for conducting convolutions in the spectral domain. ChebyNetII (He, Wei, and Wen 2022) aim to revise the Chebyshev basis by interpolation to avoid runge phenomenon, and make a significant improvement than ChebyNet. While these polynomial-based method are limited as they are scaleless of graph spectral signals.

4.3 Wavelet GNNs

Wavelet GNN via spectral graph theory is proposed in (Hammond, Vandergheynst, and Gribonval 2011). This work defines the wavelet transform on graph and inspire many works. GWNN (Xu et al. 2019a) applies graph wavelet transform to design a spectral GNN. GWGR (Cui et al. 2020b) use the LSTM structure with wavelet transform to capture network-scale and traffic prediction. GraphWave (Donnat et al. 2018) employs the heat kernel wavelet to learn a low-dimensional embedding. WNNs (Hy and Kondor 2022) propose a Multiresolution Matrix Factorization(MMF) learning algorithm to learn wavelet basis. Research (Opolka et al. 2022) constructs Gaussian process model using spectral graph wavelets to capture different scales information of neighborhood. However, the wavelets methods mentioned above are mainly use Chebyshev polynomials to do approximation, and limit the ability of wavelets basis. So we propose a novel way to use scaling function of wavelets to capture the different scale information of graph spectral signals.

5 Experiments

In this section, we conduct three main experiments. 1) We evaluate *WaveNet*'s ability to learn complex filters, such as comb, band-rejection and low-band-pass filters, etc. 2) We conduct node classification experiments to evaluate *WaveNet*'s capability to reconstruct spectral filter on real world datasets. 3) Additionally, we evaluate the effectiveness of *WaveNet* in capturing high-frequency spectral signals. At last we visualize the filters' shape learned by *WaveNet*.

5.1 Learning Complex Filters on Images Graph with WaveNet

Data: We follow the data collection in (He et al. 2021), 50 real images with the resolution of 100×100 from the Image Processing Toolbox in Matlab. Each image can be considered as a 2D regular 4-neighborhood grid graph, each image's pixel is a node in the graph and the pixel intensity translates to a signal vector $\boldsymbol{x}_i \in \mathbb{R}^{10^4}$. Since each image has the same size and node position, 50 images share the same adjacent matrix $\mathbf{A} \in \mathbb{R}^{10^4 \times 10^4}$, and we preproccess the eigenvalues decomposition for convenience.

Setup: To verify the performance of *WaveNet* fitting complex filter, we set five filters illustrated in Table 1 We choose five polynomial-based spectral GNNs as baselines: GCN (Kipf and Welling 2017), ChebyNet (Defferrard, Bresson, and Vandergheynst 2016), GPR-GNN (Chien et al. 2021), BernNet (He et al. 2021), and JacobiConv (Wang and Zhang 2022). Training data is 50 images' pixel intensity matrix mentioned above, and training goal is minimizing the sum of squared error between model prediction $\hat{\boldsymbol{x}}_i$ and pre-filtered pixel intensity vector $\tilde{\boldsymbol{x}}_i$. More detail can be found in Appendix A.3.

Results: We use the sum of squared error and R^2 score as evaluation metrics. In training, we found that our model usually reached early stopping within 500 epochs earlier than polynomial-based models, which validates that our model has a stronger learning ability. Table 1 shows the node regression results. Specially, we found that *WaveNet* excels at fitting complex filters (e.g., band or comb filters) compared to simpler ones (e.g., low or high filters). This observation validates wavelets are inherently designed to handle high-frequency component signals. In addition, we visualize the graph filters learned by BernNet and *WaveNet* in Figure 3, providing further validation of our claims. These results are based on the identical quantity of bases, with the number of bases set at 10 in both models. And *WaveNet* has achieved better filter fitting performance than BernNet. As the number of wavelet bases increases, *WaveNet* fits the ground truth more precisely, visualization results show in Appendix A.1, demonstrating its superior learning capability over polynomial-based models.

5.2 Node classification

Data: We conduct node classification task on real-world datasets following (He et al. 2021). Including three citation network Cora, CiteSeer and PubMed (Sen et al. 2008; Yang, Cohen, and Salakhudinov 2016) and the Amazon co-purchase graph Computers and Photo (McAuley et al. 2015).We also include the Wikipedia graph Chameleon and Squirrel (Rozemberczki, Allen, and Sarkar 2021), the Actor co-occurrence graph is attained from (Pei et al. 2020).

Table 1: Node regression results, mean of the sum of squared error(lower is better) & R^2 score(higher is better).

	Low-pass	High-pass	Band-pass	Band-rejection	Comb
	$\exp(-10\lambda^2)$	$1-\exp(-10\lambda^2)$	$\exp(-10(\lambda-1)^2)$	$1-\exp(-10(\lambda-1)^2)$	$\lvert\sin(\pi\lambda)\rvert$
GCN	3.4799(.9872)	67.6635(.2364)	25.8755(.1148)	21.0747(.9438)	50.5120(.2977)
GPR-GNN	0.4169(.9984)	0.0943(.9986)	3.5121(.8551)	3.7917(.9905)	4.6549(.9311)
ChebNet	0.8220(.9973)	0.7867(.9903)	2.2722(.9104)	2.5296(.9934)	4.0735(.9447)
BernNet	0.0314(.9999)	0.0113(.9999)	0.0411(.9984)	0.9313(.9973)	0.9982(.9868)
JacobiConv	**0.0003(.9999)**	**0.0064(.9999)**	0.0213(.9999)	0.0156(.9999)	0.2933(.9995)
WaveNet	0.0090(.9999)	0.0088(.9999)	**0.0076(.9997)**	**0.0074(.9999)**	**0.0813(.9996)**

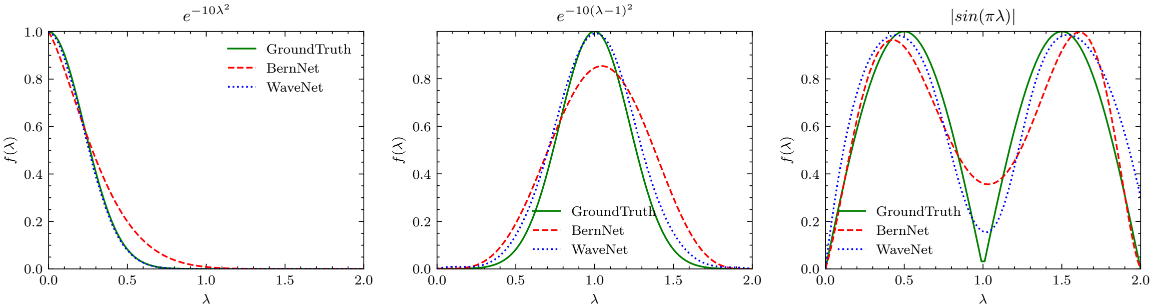

Figure 3: An illustration of learned filters by BernNet and *WaveNet*

And all the datasets are processed by the Pytorch Geometric library (Fey and Lenssen 2019) into a unified processing format. Datasets' detail information can be found in Appendix A.2.

Setup: We adopt the randomly splitting the node set into train/validation/test set with ratio 60%/20%/20%, and conduct full-supervised node classification task with each baseline model. For fairness, we also use the same evaluate setting as fowling (He et al. 2021): we generate 10 random splits by random seeds and evaluate all models on the same splits, and report the average metric for each model. The baseline models are: MLP, GCN, GAT (Veličković et al. 2018), GPR-GNN, BernNet, ChebyNetII (He, Wei, and Wen 2022), SplineCNN (Fey et al. 2018), ACMII-GCN (Luan et al. 2021) and GWNN (Xu et al. 2019a). We evaluate each baseline models through public github repository, and each hyperparameters are set as same as their official researches. Specifically, GWNN occurs gradient explosion when training since the official repository use the older version tensorflow package which makes challenging to reproduce. For other models, we use the corresponding Pytorch Geometric library implementations. More details are showing in the Appendix A.3

Results: *WaveNet* achieves the best performance on Computers and Photo datasets among the mentioned models. Although the performance is not the best on citation networks, *WaveNet* still attains superior performance. The performance on heterogeneous graphs (e.g., Chameleon and Squirrel) achieves approximately 1.23% and 5.25% accuracy improvement. These results demonstrate that *WaveNet* has an excellent performance in handling complex graph spectral signals. Regarding the poor performance on Actor dataset, we have more discussions in Appendix A.4.

5.3 High-frequency Spectral Signal Node classification

Our synthetic toy graph is based on cSBM (Deshpande et al. 2018). We employ cSBM($\phi = -0.25$, number of nodes is 50, features dimension is 6) to generate a graph $\mathcal{G}$ with its Laplacian matrix $\mathbf{L}$. And then we define pre-determined spectral signals with high-frequency to reconstruct nodes labels. We choose $\sin(10\cdot\lambda)+1$ and $\sin(10\cdot\lambda)\cdot e^{-\lambda}+1$ as the ground truth signals to evaluate whether *WaveNet* capture the fast changes in the graph. The baseline models are BernNet and ChebyNetII which are representative polynomial-based model in Spectral GNNs domain. For fairness, the order of polynomial filter $K = 10$ and the number of wavelet bases as well.

Results: The results presented in Table 3 indicate that the wavelet-based filtering approach significantly outperforms the polynomial basis method. Figure 4 illustrates that our proposed model, *WaveNet*, demonstrates the ability to capture the scale of graph spectral and exhibits sensitivity to rapidly changing signals. It shows that *WaveNet* is particularly effective when dealing with high-frequency spectral graph signals. This experiment demonstrates that polynomial-based models have limited ability to fit complex filters, whereas *WaveNet* leverages wavelet-based methods to overcome the constraints of polynomial-based models and effectively capture rapidly changing components in the graph spectral.

5.4 Visualize the Filters

In this section, we further explore polynomial-based and wavelet-based spectral domain filtering methods. By visualizing the scaling function coefficients learned by our model

Table 2: Results on real world benchmark datasets: Mean accuracy (%) ± 95% confidence interval. Grey number means gradient explosion when training.

	Cora	CiteSeer	PubMed	Computers	Photo	Chameleon	Squirrel	Actor
				Spectral GNNs				
GCN	87.14$_{\pm 1.01}$	79.86$_{\pm 0.67}$	86.74$_{\pm 0.27}$	83.32$_{\pm 0.33}$	88.26$_{\pm 0.73}$	59.61$_{\pm 2.21}$	46.78$_{\pm 0.87}$	33.23$_{\pm 1.16}$
GAT	88.03$_{\pm 0.79}$	80.52$_{\pm 0.71}$	87.04$_{\pm 0.24}$	83.32$_{\pm 0.39}$	90.94$_{\pm 0.68}$	63.13$_{\pm 1.93}$	44.49$_{\pm 0.88}$	33.93$_{\pm 2.47}$
MLP	76.96$_{\pm 0.95}$	76.58$_{\pm 0.88}$	85.94$_{\pm 0.22}$	82.85$_{\pm 0.38}$	84.72$_{\pm 0.34}$	46.85$_{\pm 1.51}$	31.03$_{\pm 1.18}$	40.19$_{\pm 0.56}$
GPR-GNN	88.57$_{\pm 0.69}$	80.12$_{\pm 0.83}$	88.46$_{\pm 0.33}$	86.85$_{\pm 0.25}$	93.85$_{\pm 0.28}$	67.28$_{\pm 1.09}$	50.15$_{\pm 1.92}$	39.92$_{\pm 0.67}$
BernNet	88.52$_{\pm 0.95}$	80.09$_{\pm 0.79}$	88.48$_{\pm 0.41}$	87.64$_{\pm 0.44}$	93.63$_{\pm 0.35}$	68.29$_{\pm 1.58}$	51.35$_{\pm 0.73}$	41.79$_{\pm 1.01}$
ChebyNetII	88.71$_{\pm 0.93}$	80.53$_{\pm 0.79}$	88.93$_{\pm 0.29}$	91.27$_{\pm 0.45}$	94.06$_{\pm 0.37}$	71.37$_{\pm 1.01}$	57.72$_{\pm 0.59}$	41.75$_{\pm 1.07}$
SplineCNN	86.90$_{\pm 0.47}$	78.43$_{\pm 0.64}$	86.73$_{\pm 0.39}$	82.08$_{\pm 4.72}$	93.66$_{\pm 0.41}$	51.90$_{\pm 1.35}$	37.75$_{\pm 0.92}$	35.11$_{\pm 0.82}$
ACMII-GCN	**89.00**$_{\pm \mathbf{0.72}}$	**81.79**$_{\pm \mathbf{0.95}}$	**90.74**$_{\pm \mathbf{0.50}}$	-	-	68.38$_{\pm 1.36}$	54.53$_{\pm 2.09}$	**41.84**$_{\pm \mathbf{1.15}}$
				Wavelet GNNs				
GWNN	86.21$_{\pm 0.00}$	78.99$_{\pm 0.00}$	81.65$_{\pm 0.00}$	84.77$_{\pm 0.00}$	86.18$_{\pm 0.00}$	20.79$_{\pm 0.00}$	19.12$_{\pm 0.00}$	0.00$_{\pm 0.00}$
WaveNet(ours)	88.46$_{\pm 0.51}$	80.22$_{\pm 0.71}$	90.26$_{\pm 0.22}$	**92.06**$_{\pm \mathbf{0.33}}$	**94.42**$_{\pm \mathbf{0.31}}$	**72.60**$_{\pm \mathbf{1.18}}$	**62.97**$_{\pm \mathbf{0.62}}$	39.69$_{\pm 1.54}$

Table 3: Node classification task in 50 nodes synthetic toy graph. Labels are generated by two high-frequency spectral signals . Mean accuracy (%) ± 95% confidence interval.

	$sin(10\lambda)+1$	$sin(10\lambda)e^{-\lambda}+1$
BernNet	63.57$_{\pm 5.00}$	50.91$_{\pm 7.27}$
ChebyNetII	70.71$_{\pm 10.00}$	47.27$_{\pm 7.27}$
WaveNet	**74.29**$_{\pm \mathbf{5.71}}$	**68.18**$_{\pm \mathbf{9.09}}$

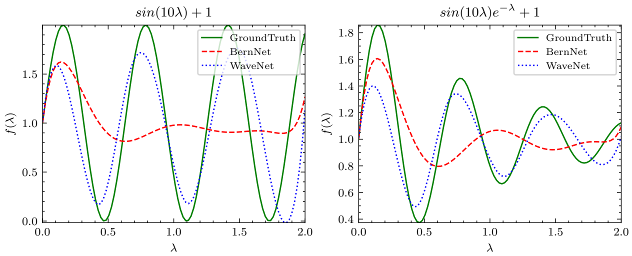

Figure 4: Illustration of high-frequency spectral signals learned by BerNet and *WaveNet*.

and their corresponding shapes as shown in Figure 5, we gain a better understanding of their behavior. By visualizing the learned filters, we can observe high-frequency variations occurring locally across different datasets. This observation further reinforces our initial proposition: 1) real world datasets' filter is more complex and exist local mutations; 2) since polynomial-based models are scaleless, they are struggle to capture this dramatic changes. However, these high-frequency variations can be effectively captured in a comprehensive manner by using a wavelet-based approach. The visualizations provide valuable insights into the superiority of wavelet-based methods in modeling and representing local variations across datasets with varying degrees of complexity. More datasets' filter shape can be found in Appendix A.5

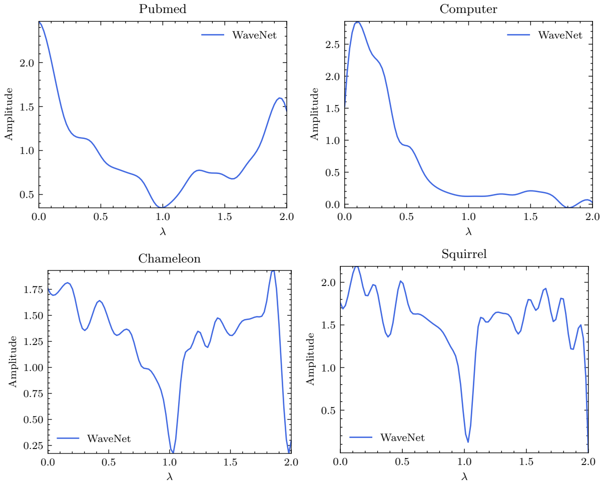

Figure 5: Illustration of filters learned from real-world datasets by *WaveNet*.

6 Conclusion

Our work breaks the dominance of polynomial-based approaches in Spectral GNNs and introduces a novel perspective for researchers. We employ MRA on spectral graph signal, and demonstrate wavelets' strong capacity on high-frequency signals fitting. Then we utilize scaling function to reconstruct the spectral signals in graph. Through our experiments, First, we validate the effectiveness of the wavelet-based approach on learning complex filters by image graph experiment. Second, the node classification results demonstrate the superiority of wavelet-based methods over traditional polynomial-based methods. Third, synthetic toy experiment further validate *WaveNet* has a better capability to capture fine components compared to polynomial methods. Last, we visualize the learned filters. Future research will aim to tackle computational complexity and try to integrate alternative wavelet bases like Daubechies wavelet family into *WaveNet*. We also want to explore wavelets' potential in small molecular graphs.

References

Banerjee, A. 2008. *The spectrum of the graph Laplacian as a tool for analyzing structure and evolution of networks*. Ph.D. thesis, Verlag nicht ermittelbar.

Bo, D.; Shi, C.; Wang, L.; and Liao, R. 2023. Specformer: Spectral Graph Neural Networks Meet Transformers. In *ICLR*. OpenReview.net.

Bogaerts, T.; Masegosa, A. D.; Angarita-Zapata, J. S.; Onieva, E.; and Hellinckx, P. 2020. A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. *Transportation Research Part C: Emerging Technologies*.

Boggess, A.; and Narcowich, F. J. 2015. *A first course in wavelets with Fourier analysis*. John Wiley & Sons.

Chien, E.; Peng, J.; Li, P.; and Milenkovic, O. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. In *ICLR*.

Cui, Z.; Ke, R.; Pu, Z.; Ma, X.; and Wang, Y. 2020a. Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction. *Transportation Research Part C: Emerging Technologies*.

Cui, Z.; Ke, R.; Pu, Z.; Ma, X.; and Wang, Y. 2020b. Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction. *Transportation Research Part C: Emerging Technologies*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, 3844–3852.

Deshpande, Y.; Sen, S.; Montanari, A.; and Mossel, E. 2018. Contextual Stochastic Block Models. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Do, V.; Corbett-Davies, S.; Atif, J.; and Usunier, N. 2022. Online certification of preference-based fairness for personalized recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Donnat, C.; Zitnik, M.; Hallac, D.; and Leskovec, J. 2018. Learning structural node embeddings via diffusion wavelets. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining*, 1320–1329.

Feng, J.; Chen, Y.; Li, F.; Sarkar, A.; and Zhang, M. 2022. How powerful are k-hop message passing graph neural networks. *Advances in Neural Information Processing Systems*.

Fey, M.; and Lenssen, J. E. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR*.

Fey, M.; Lenssen, J. E.; Weichert, F.; and Müller, H. 2018. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Gao, C.; Zheng, Y.; Li, N.; Li, Y.; Qin, Y.; Piao, J.; Quan, Y.; Chang, J.; Jin, D.; He, X.; and Li, Y. 2022. A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions. *ACM Transactions on Recommender Systems (TORS)*.

Guo, T.; Zhang, T.; Lim, E.; Lopez-Benitez, M.; Ma, F.; and Yu, L. 2022. A review of wavelet analysis and its applications: Challenges and opportunities. *IEEE Access*.

Haar, A. 1909. *Zur theorie der orthogonalen funktionensysteme*. Georg-August-Universitat, Gottingen.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *NeurIPS*.

Hammond, D. K.; Vandergheynst, P.; and Gribonval, R. 2011. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*.

He, M.; Wei, Z.; Huang, Z.; and Xu, H. 2021. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. In *NeurIPS*.

He, M.; Wei, Z.; and Wen, J.-R. 2022. Convolutional Neural Networks on Graphs with Chebyshev Approximation, Revisited. In *NeurIPS*.

Hewitt, E.; and Hewitt, R. E. 1979. The Gibbs-Wilbraham phenomenon: an episode in Fourier analysis. *Archive for history of Exact Sciences*.

Hy, T. S.; and Kondor, R. 2022. Multiresolution Matrix Factorization and Wavelet Networks on Graphs. In *Proceedings of Topological, Algebraic, and Geometric Learning Workshops 2022*, 172–182. PMLR.

Jiang, D.; Wu, Z.; Hsieh, C.-Y.; Chen, G.; Liao, B.; Wang, Z.; Shen, C.; Cao, D.; Wu, J.; and Hou, T. 2021. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *Journal of cheminformatics*.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then propagate: Graph neural networks meet personalized pagerank. In *ICLR*.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*.

Luan, S.; Hua, C.; Lu, Q.; Zhu, J.; Zhao, M.; Zhang, S.; Chang, X.-W.; and Precup, D. 2021. Is heterophily a real nightmare for graph neural networks to do node classification? *arXiv preprint arXiv:2109.05641*.

Mallat, S. G. 1989. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*.

McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*.

Opolka, F.; Zhi, Y.-C.; Lió, P.; and Dong, X. 2022. Adaptive Gaussian Processes on Graphs via Spectral Graph Wavelets. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, 4818–4834. PMLR.

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-gcn: Geometric graph convolutional networks. In *ICLR*.

Powell, M. J. D. 1981. *Approximation theory and methods*. Cambridge university press.

Rong, Y.; Bian, Y.; Xu, T.; Xie, W.; Wei, Y.; Huang, W.; and Huang, J. 2020. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems(NeurIPS)*.

Rozemberczki, B.; Allen, C.; and Sarkar, R. 2021. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2): cnab014.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Shuman, D. I.; Narang, S. K.; Frossard, P.; Ortega, A.; and Vandergheynst, P. 2013. The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains. *IEEE Signal Process. Mag.*

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.

Wang, X.; and Zhang, M. 2022. How Powerful are Spectral Graph Neural Networks. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR.

Wu, Z.; Jain, P.; Wright, M.; Mirhoseini, A.; Gonzalez, J. E.; and Stoica, I. 2021. Representing Long-Range Context for Graph Neural Networks with Global Attention. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Xu, B.; Shen, H.; Cao, Q.; Qiu, Y.; and Cheng, X. 2019a. Graph Wavelet Neural Network. In *ICLR*. OpenReview.net.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019b. How powerful are graph neural networks? In *ICLR*.

Yan, Y.; Hashemi, M.; Swersky, K.; Yang, Y.; and Koutra, D. 2022. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. In *2022 IEEE International Conference on Data Mining (ICDM)*.

Yang, Z.; Cohen, W.; and Salakhudinov, R. 2016. Revisiting semi-supervised learning with graph embeddings. In *ICML*, 40–48. PMLR.

Yang, Z.-Y.; Yang, Z.-J.; Dong, J.; Wang, L.-L.; Zhang, L.-X.; Ding, J.-J.; Ding, X.-Q.; Lu, A.-P.; Hou, T.-J.; and Cao, D.-S. 2019. Structural analysis and identification of colloidal aggregators in drug discovery. *Journal of chemical information and modeling*.

Zhang, J.; Zhang, H.; Xia, C.; and Sun, L. 2020. Graph-Bert: Only Attention is Needed for Learning Graph Representations. *arXiv preprint arXiv:2001.05140*.

Zhu, J.; Rossi, R. A.; Rao, A.; Mai, T.; Lipka, N.; Ahmed, N. K.; and Koutra, D. 2021. Graph neural networks with heterophily. In *Proceedings of the AAAI conference on artificial intelligence*.

Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

A Appendix

A.1 Visualization Results about Node Regression

We further explore the influence about node regression with the number of wavelet bases changes. We conduct *WaveNet* to learn more complex filters and the visualization results shows in Figure 6.

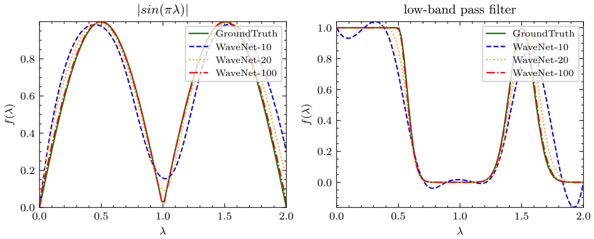

Figure 6: Illustration of complex filters learned by *WaveNet* with the number of wavelet bases changing.

A.2 Datasets Detail

All the datasets can be found in Pytorch Geometric library, and detail information show in Table 4.

Table 4: Dataset statistics.

	Nodes	Edges	Features	Classes
Cora	2708	5278	1433	7
CiteSeer	3327	4552	3703	6
PubMed	19717	44324	500	5
Computers	13752	245861	767	10
Photo	7650	119081	745	8
Chameleon	2277	31371	2325	5
Squirrel	5201	198353	2089	5
Actor	7600	26659	932	5

A.3 The WaveNet Architecture Detail

In this section, we will introduce *WaveNet*'s architecture. Let $\mathbf{X}^{(0)} \in \mathbb{R}^{n\times d}$ be an input matrix of graph node feature. We employ linear layer to map the input feature to hidden space, which can be easily describe as following:

$$\mathbf{X}^{l} = \sigma(\mathbf{W}^{l}\mathbf{X}^{(l-1)} + \mathbf{B}^{l-1}). \tag{18}$$

Since the Haar wavelet is not a continuous function in the interval of $[0, 2]$, we pre-compute an eigendecomposition of the Laplacian matrix, and conduct scaling function of wavelets to reconstruct the spectral signal of graph. And the message propagation process is described as following:

$$\boldsymbol{y} = \text{sofmax}(\tilde{\mathbf{L}}\mathbf{X}^{l}). \tag{19}$$

WaveNet Setup Detail When Learning Complex Filter: To ensure fairness, all baseline models are using two convolutional units and a linear output layer. The convolutional units' hidden dimension is $(32, 64)$ respectively. The number of polynomial and wavelet bases is kept at 10. The maximum number of training epochs is set to 2000, and the early stopping threshold is set to 200 epochs if the loss does not decrease. The Adam optimizer is selected and learning rate is set to 0.01 for all models, loss function is the sum of squared error. To force the model to learn the correct coefficients of the ground truth filters, we remove all linear layers from the model, retaining only the propagation and convolutional layers. Although this setup increases the squared error, it helps the model learn the correct coefficients.

WaveNet Setup Detail When Node Classification: For fairness, we employ *WaveNet* the same architecture using 2-layer linear with 64 hidden units on the graph nodes feature matrix $\mathbf{X}$. After we get the hidden dimensions representation of nodes feature, the propagation matrix $\mathbf{P}$ will apply on hidden feature matrix $\mathbf{X}^{l}$. Last, we utilize a MLP layer to do classification. And this kind of network structure is mentioned in (He et al. 2021). We follow the same framework as previous work employed. We use the micro-F1 score with a 95% confidence interval as the evaluation metric. The relevant results are summarized in Table 2.

A.4 Analysis of the Actor dataset

Table 5: Eigenvalues distribution of Actor dataset.

Eigenvalue Interval	Number of Eigenvalue
[0.0,0.2)	63
[0.2,0.4)	439
[0.4,0.6)	796
[0.6,0.8)	951
[0.8,1.0)	2116
[1.0,1.2)	2081
[1.2,1.4)	1030
[1.4,1.6)	865
[1.6,1.8)	373
[1.8,2.0]	35

Previous works show that the filter of **Actor** is a line-like filter which is called all-pass-alike filter (He et al. 2021; Chien et al. 2021; Bo et al. 2023). Theoretically, wavelets basis can sensitively capture the fluctuation of spectral signals. And the true filter of the **Actor** dataset is indeed high-frequency, then *WaveNet* can effectively learn an appropriate filter to handle it. While our model, similar to the current state-of-the-art model, also failed to achieve significant improvement in terms of accuracy. Therefore it is essential to analyze the underlying reasons for this phenomenon.

As demonstrated in Table 5, we obtained the eigenvalue distribution of the Laplacian matrix. It is evident that the distribution is predominantly concentrated around $\lambda = 1$. Additionally, out of the total eigenvalues, there are 1135 eigenvalues that equal 1, which accounts for approximately 15% of the eigenvalue spectral. This observation implies that approximately 15% of the nodes are connected to the same neighbors in the graph (Banerjee 2008), suggesting that the **Actor** dataset represents a highly isomorphic graph. We believe that the strong isomorphism of the graph is one of the main factors contributing to the current poor prediction performance on the **Actor** dataset.

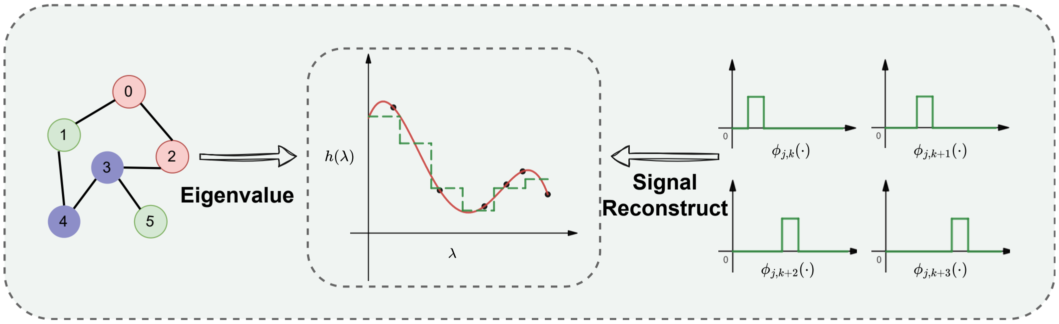

Figure 7: An illustration architecture of WaveNet. The signal reconstruct process is based on wavelet bases and we utilize the Haar wavelet to filter on graph data.

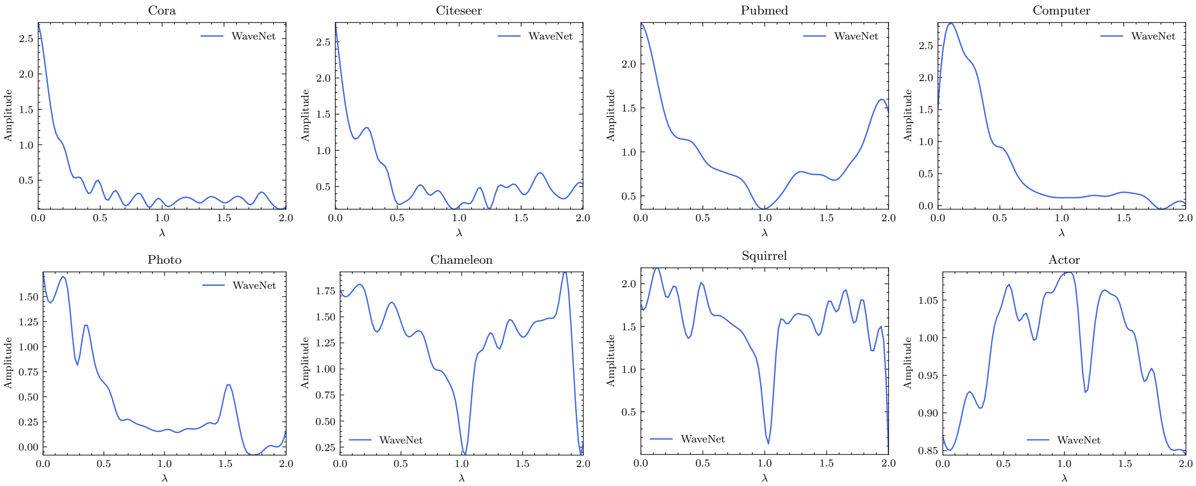

Figure 8: Illustration of filters learned from real-world datasets by *WaveNet*.

Considering our model *WaveNet*, is not specifically designed to address graph isomorphism. Therefore, the performance of *WaveNet* is acceptable. To verify whether our model can effectively capture the high-frequency components of graph data, we conduct experiments on artificially synthesized spectral graph data with high-frequency spectral signals.

A.5 Filter Learned by WaveNet

Figure 8 plots the filters learnt from real-world datasets by *WaveNet*. And most filters exhibit localized mutation properties