When Spectral Domain Meets Spatial Domain in Graph Neural Networks

Balcilar Muhammet [1] **Renton Guillaume** [1] **Héroux Pierre** [1] **Gaüzère Benoit** [2] **Adam Sébastien** [1] **Honeine Paul** [1]

Abstract

Convolutional Graph Neural Networks (ConvGNNs) are designed either in the spectral domain or in the spatial domain. In this paper, we provide a theoretical framework to analyze these neural networks, by deriving some equivalence of the graph convolution processes, regardless if they are designed in the spatial or the spectral domain. We demonstrate the relevance of the proposed framework by providing a spectral analysis of the most popular ConvGNNs (ChebNet, CayleyNet, GCN and Graph Attention Networks), which allows to explain their performance and shows their limits.

1. Introduction

Deep Learning with Convolutional Neural Networks (CNNs) and Recurrent Neural Networks has been having a significant impact in many machine learning applications for signal and image processing, such as image recognition (Krizhevsky et al., 2012) and speech analysis (Graves et al., 2013). These successes have mostly been achieved on sequences or images, i.e., on data defined on grid structures that allow to carry out linear algebra operations in Euclidean spaces. However, there are many domains where data cannot be trivially encoded into an Euclidean domain, but can be naturally represented as graphs, such as with social networks, molecules, knowledge graph. For this purpose, Graph Neural Networks (GNNs) have been recently proposed in the literature (Scarselli et al., 2009; Gilmer et al., 2017; Bronstein et al., 2017; Wu et al., 2019b). GNNs are Neural Networks that rely on the computation of hidden representations of nodes using information carried by the whole graph.

Convolutional GNNs (ConvGNNs) aim to mimic the simple and efficient solution provided by CNNs to extract features through a weight-sharing strategy along the presented data.

[1]LITIS Lab, University of Rouen Normandy, Rouen, France. [2]LITIS Lab, INSA Rouen, Rouen, France. Correspondence to: Balcilar Muhammet <muhammetbalcilar@gmail.com>.

ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+), Vienna, Austria.

In the literature, two strategies have been investigated to design filter kernels, either in the spectral or in the spatial domains.

Spectral-based convolution filters are defined using a graph signal processing point of view. However, despite the solid mathematical foundations borrowed from the signal processing literature, such approaches suffer from (i) the large computational burden induced by the forward/inverse graph Fourier transform, (ii) being spatially non-localized and (iii) the transferability problem, i.e., filters designed using a given graph cannot be applied on other graphs. To alleviate these issues, some approaches based on parameterization have been proposed, using B-spline (Bruna et al., 2013), Chebyshev polynomials (Defferrard et al., 2016) and Cayley polynomials (Levie et al., 2019). The second strategy is the spatial-based convolution, which aggregates nodes neighborhood information, in the same spirit as the conventional Euclidean convolution (e.g. 2D convolution in CNNs). Such convolutions have been very attractive due to their less computational complexity, their localized property and their transferability. However, their spectral behavior is not taken into account and they are generally poor on creating various frequency component signal on the output profile.

Spectral and spatial approaches are generally studied separately (Wu et al., 2019b; Chami et al., 2020). However, Message Passing Neural Networks is the first and, to the best of our knowledge, unique attempt to merge both in the same framework (Gilmer et al., 2017). They fitted some spectral approaches into their framework as a Laplacian based message passing schema. However their model was not able to generalize custom designed spectral filters as well as the effect of each convolution support in multi convolution case. The spatial-spectral connection is also mentioned in cornerstone researches in (Defferrard et al., 2016; Kipf & Welling, 2017; Levie et al., 2019) indirectly. Although their graph convolutions are spectral designed, thank to the Chebyshev and Cayley polynomials, they managed to write convolutions in spatial domain. Since the two way interchangeability (spectral-spatial) is missing in all these mentioned researches, they did not propose a spectral analysis of any graph convolutions. Researches attempted to show that a very limited number of the spatial convGNNs work as low-pass filtering (NT & Maehara, 2019; Wu et al., 2019a). (NT & Maehara, 2019) concluded that using adjacency induces

a low-pass filtering. These researches proposed a spectral analysis on single spatial approach and did not generalize it for any graph convolution.

In this paper, we provide connections between spectral and spatial domains for ConvGNNs. The main contribution of this paper is a theoretical framework that demonstrates the equivalence of convolution processes, regardless if they are designed in the spatial or the spectral domain. By investigating this proposed framework, we demonstrate its relevance by providing a spectral analysis of existing graph convolutions for four well-known ConvGNNs, known as GCN (Kipf & Welling, 2017), ChebNet (Defferrard et al., 2016), CayleyNet (Levie et al., 2019) and Graph Attention Networks (GAT) (Veličković et al., 2018).

2. Connecting Spatial and Spectral Domains

Spectral ConvGNNs rely on the spectral graph theory. For a given graph, let U be the eigenvectors matrix of its graph Laplacian L, and $\boldsymbol{\lambda}$ the vector of its eigenvalues. A graph convolution layer in spectral domain can be written as a sum of filtered signals followed by an activation function σ (e.g. `RELU`) as in (Bruna et al., 2013), namely

$$H_j^{(l+1)} = \sigma\Big(\sum_{i=1}^{f_l} U \operatorname{diag}(F_{i,j,l}) U^\top H_i^{(l)}\Big), \tag{1}$$

for all $j \in \{1, \dots, f_{l+1}\}$, where $H_j^{(l)}$ is the j-th feature vector of the l-th layer, and $F_{i,j,l}$ is the trainable weight vector. This formulation is intractable since it requires computing the graph Fourier transform and its inverse, by matrix multiplication of U and U^T. Another drawback is the filter non-transferability in multi-graph learning problems. To overcome these issues, it is often re-parameterized as

$$F_{i,j,l} = B \left[W_{i,j}^{(l,1)}, \dots, W_{i,j}^{(l,S)}\right]^\top, \tag{2}$$

where $B \in \mathbb{R}^{n \times S}$ is the parameterization matrix, n is number of nodes in the given graph, S is the desired number of convolution kernels and $W^{(l,s)}$ is the trainable matrix with index $s = 1..S$. Each column in B is designed as a function of eigenvalues, i.e., $B_{i,j} = F_j(\boldsymbol{\lambda}_i)$, such as with B-spline (Bruna et al., 2013) or polynomial (Defferrard et al., 2016).

The following theorem is the main theoretical contribution of the paper.

Theorem 1. *Spectral ConvGNN parameterized with fixed frequency profiles matrix B of entries $B_{i,j} = F_j(\boldsymbol{\lambda}_i)$, defined as*

$$H_j^{(l+1)} = \sigma\Big(\sum_{i=1}^{f_l} U \operatorname{diag}\Big(B \left[W_{i,j}^{(l,1)}, \dots, W_{i,j}^{(l,S)}\right]^\top\Big) U^\top H_i^{(l)}\Big), \tag{3}$$

is a particular case of spatial ConvGNN, defined as

$$H^{(l+1)} = \sigma\Big(\sum_s C^{(s)} H^{(l)} W^{(l,s)}\Big), \tag{4}$$

with the convolution kernel set to

$$C^{(s)} = U \operatorname{diag}(F_s(\boldsymbol{\lambda})) U^\top. \tag{5}$$

Proof. First, let us expand the matrix B and rewrite it as the sum of its columns, denoted $F_1(\boldsymbol{\lambda}), \dots, F_S(\boldsymbol{\lambda}) \in \mathbb{R}^n$:

$$H_j^{(l+1)} = \sigma\Big(\sum_{i=1}^{f_l} U \operatorname{diag}\Big(\sum_{s=1}^{S} W_{i,j}^{(l,s)} F_s(\boldsymbol{\lambda})\Big) U^\top H_i^{(l)}\Big). \tag{6}$$

We distribute U and $U^\top$ over the inner summation, and then take out the scalars $W_{i,j}^{(l,s)}$ of the diag operator:

$$H_j^{(l+1)} = \sigma\Big(\sum_{s=1}^{S} \sum_{i=1}^{f_l} W_{i,j}^{(l,s)} U \operatorname{diag}(F_s(\boldsymbol{\lambda})) U^\top H_i^{(l)}\Big). \tag{7}$$

Let us define a convolution operator $C^{(s)} \in \mathbb{R}^{n \times n}$ as:

$$C^{(s)} = U \operatorname{diag}(F_s(\boldsymbol{\lambda})) U^\top. \tag{8}$$

Using (7) and (8), we have thus:

$$H_j^{(l+1)} = \sigma\Big(\sum_{i=1}^{f_l} \sum_{s=1}^{S} W_{i,j}^{(l,s)} C^{(s)} H_i^{(l)}\Big). \tag{9}$$

Then, each term of the sum over s corresponds to a matrix $H^{(l+1)} \in \mathbb{R}^{n \times f_{l+1}}$, with

$$H^{(l+1)} = \sigma\Big(\sum_{s=1}^{S} C^{(s)} H^{(l)} W^{(l,s)}\Big), \tag{10}$$

where $H^{(l)} = [H_1^{(l)}, \dots, H_{f_l}^{(l)}]$. Therefore, (3) corresponds to (4) with $C^{(s)}$ defined as (5). □

Corollary 1.1. *The frequency profile of any given graph convolution kernel $C^{(s)}$ can be defined in spectral domain by the vector where diag^{-1} returns the diagonal of given matrix as a vector.*

$$F_s(\boldsymbol{\lambda}) = \operatorname{diag}^{-1}(U^\top C^{(s)} U). \tag{11}$$

Proof. By using (5) from Theorem 1, we can obtain a spatial convolution kernel $C^{(s)}$ whose frequency profile is $F_s(\boldsymbol{\lambda})$. Since the eigenvector matrix is orthonormal (i.e., $U^{-1} = U^\top$), we can extract $F_s(\boldsymbol{\lambda})$, which yields (11). □

We denote the matrix $F_s = U^\top C^{(s)} U$ as the full frequency profile of the convolution kernel $C^{(s)}$, and $F_s(\boldsymbol{\lambda}) = \operatorname{diag}(F_s)$ as the standard frequency profile of the convolution kernel. The full frequency profile includes all eigenvector-to-eigenvector pairs contributions. Standard frequency profile just includes each eigenvector's self-contribution.

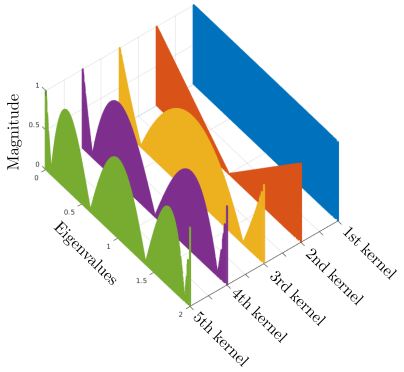


Figure 1. Standard frequency profiles of the first 5 Chebyshev convolution kernels.

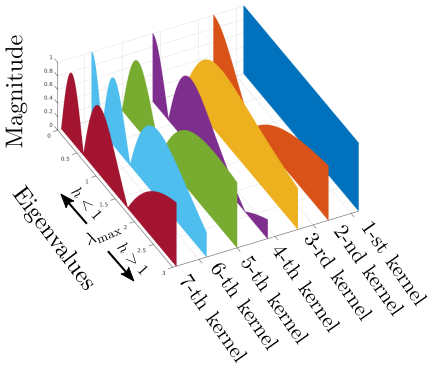


Figure 2. Standard frequency profiles of the first 7 CayleyNet convolution kernels.

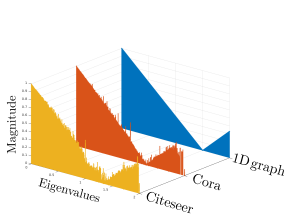
3. Spectral Analysis of Graph Convolutions

To show the frequency profiles of some well-known graph convolutions, we used a 1D signal encoded as a regular circular line graph with 1001 nodes and the Cora and Citeseer reference datasets, which consist of one single graph with respectively 2708 and 3327 nodes (Yang et al., 2016). Basically, each node of these graphs is labeled by a vector, and edges are unlabeled and undirected.

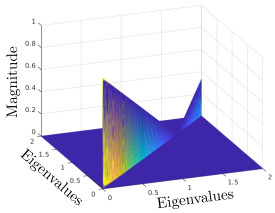
ChebNet: The first two Chebyshev kernels are $C^{(1)} = I$ and $C^{(2)} = 2L/\lambda_{\max} - I$ and the remaining ones are defined by $C^{(k)} = 2C^{(2)}C^{(k-1)} - C^{(k-2)}$ (Defferrard et al., 2016), where I refers identity matrix. Corollary 1.1 gives their frequency profiles. As shown in Appendix A, the first two kernel frequency profiles of ChebNet are $F_1(\boldsymbol{\lambda}) = \mathbf{1}$ and $F_2(\boldsymbol{\lambda}) = 2\boldsymbol{\lambda}/\lambda_{\max} - \mathbf{1}$, where $\mathbf{1}$ is the vector of ones. Since $\lambda_{\max} = 2$ for all three graphs, we get $F_2(\boldsymbol{\lambda}) = \boldsymbol{\lambda} - \mathbf{1}$. The third one and following kernel frequency profiles can also be computed using $F_k(\boldsymbol{\lambda}) = 2F_2(\boldsymbol{\lambda})F_{k-1}(\boldsymbol{\lambda}) - F_{k-2}(\boldsymbol{\lambda})$, leading to $F_3(\boldsymbol{\lambda}) = 2\boldsymbol{\lambda}^2 - 4\boldsymbol{\lambda} + \mathbf{1}$ for the third kernel. The resulting 5 frequency profiles are shown in Figure 1 (in absolute value). Since the full frequency profiles consist of zeros outside the diagonal, they are not illustrated.

Analyzing these frequency profiles, one can argue that the convolutions mostly cover the spectrum. However, none of the kernels focuses on some certain parts of the spectrum. As an example, the second kernel is roughly a band-stop filter, while the third one passes very high, very low and middle bands, but stops almost first and third quarter of the spectrum. Therefore, if the relation between input-output pairs can be figured out by just a low-pass, high-pass or some specific band-pass filter, a high number of convolution kernels is needed. However, only 2 or 3 kernels are generally used in the literature (Defferrard et al., 2016).

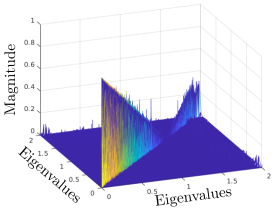
CayleyNet: In (Levie et al., 2019), CayleyNet is parameterized with $F_{i,j,l} = [g_{i,j,l}(\lambda_1, h), ..., g_{i,j,l}(\lambda_n, h)]^\top$, where



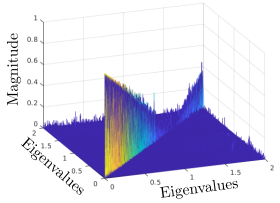
(a) Standard frequency profiles



(b) Full frequency profile on 1D regular line graph



(c) Full freq. profile on Cora



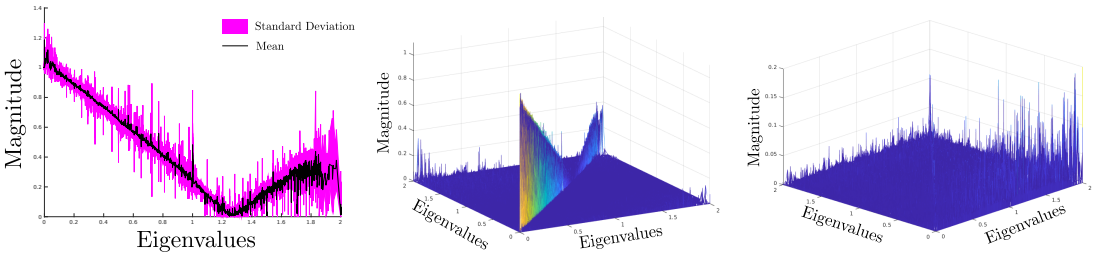
(d) Full freq. profile on Citeseer

Figure 3. Frequency profiles of GCN on different graphs.

h is a scale parameter to be learned, λ_n is the n-th eigenvalue, and g is a spectral filter function defined as

$$g(\lambda, h) = c_0 + 2Re\left(\sum_{k=1}^{r} c_k \left(\frac{h\lambda - \mathbf{i}}{h\lambda + \mathbf{i}}\right)^k\right), \qquad (12)$$

where $\mathbf{i}^2 = -1$, $Re(\cdot)$ is the real part, c_0 is a real trainable coefficient, and c_k are the complex trainable coefficients. As proven in Appendix B, CayleyNet can be defined through the frequency profile matrix B to be used in (3). Using this representation, CayletNet can be seen as multi-kernel convolutions with real-valued trainable coefficients. According to this analysis, CayleyNet uses $2r + 1$ graph convolution kernels, with r being the number of complex coefficients (Levie et al., 2019). The first 7 kernel's frequency profiles are illustrated in Figure 2. The scale parameter h affects the x-axis scaling but does not change the global shape. Learning the scaling of eigenvalues may seem advantageous. However, it induces extra computational cost. In addition, similarly to



(a) Standard frequency profile (b) Mean of full frequency profile (c) Std dev. of full frequency profile

Figure 4. Frequency profiles of randomly generated 250 GAT convolutions using Cora graph.

ChebNet, CayleyNet does not have any band specific convolutions, even when considering different scaling factors.

GCN: GCN (Kipf & Welling, 2017) uses the single kernel

$$C = \widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}, \tag{13}$$

where $\widetilde{D}_{i,i} = \sum_j \widetilde{A}_{i,j}$ and $\widetilde{A} = (A + I)$ with A being the the adjacency matrix. The theoretical analysis of frequency profiles of GCN convolution is carried out in Appendix C. It shows that GCN frequency profile can be approximated according to $F(\boldsymbol{\lambda}) \approx \mathbf{1} - \boldsymbol{\lambda}\overline{d}/(\overline{d}+1)$, where $\overline{d}$ is the average node degree. Theoretically, if all nodes degree are different, standard frequency profile will not be smooth and will include some perturbations. In addition, full frequency profile will be composed of non-zero components.

The spectral representation of GCN's convolution matrix can be back-calculated using Corollary 1.1. This result leads to the frequency profiles illustrated in Figure 3 for the three different graphs. The three standard frequency profiles have almost the same low-pass filter shape corresponding to a function composed of a decreasing part on the three first quarters of the eigenvalues range, followed by an increasing part on the remaining range. This observation is coherent with the theoretical analysis in Appendix C.

Since GCN frequency profile does not cover the whole spectrum, such an approach is not able to learn relations that can be represented by high-pass or band-pass filtering. Hence, even though it gives very good results on a single graph node classification problem in (Kipf & Welling, 2017), it may fail for problems where discriminant information lies in particular frequency bands.

GAT: GAT (Veličković et al., 2018) uses multi-attention weights (denoted as trainable convolution kernels), with

$$C_{i,j}^{(l,s)} = \texttt{softmax}_j\left(\sigma(\mathbf{a}[\mathbf{W}H_i^{(l)}, \mathbf{W}H_j^{(l)}])\right), \tag{14}$$

where two linear transformations are considered by elements of general trainable parameter set $W^{(l,s)} = \{\mathbf{a}, \mathbf{W}\}$ in (4), with $\mathbf{a}$ being a weight vector, and $\texttt{softmax}_j$ is the normalized exponential function that uses all neighbors of i-th node to normalize edge of i-th to j-th node.

Since GAT relies on trainable convolutions kernels and differs for each layer, frequency profiles cannot be directly computed similarly to GCN or ChebNet. Thus, we perform simulations and evaluate the potential kernels of attention mechanism for given graphs in first layer. Hence, we show the frequency profiles of those simulated potential kernels.

To show the potential output of GATs on the Cora graph (1433 features for each node), we produce 250 random pairs of $\mathbf{W} \in R^{1433\times 8}$ and $\mathbf{a} \in R^{16\times 1}$, corresponding to the first layer trained by GATs. The σ function in (14) is a `LeakyReLU` activation with a 0.2 negative slope as in (Veličković et al., 2018). The mean and standard deviation of the frequency profiles for these simulated GAT kernels are shown in Figure 4. As one can see, the mean standard frequency profile has a similar shape as those of GCN (Figure 3). However, variations on the frequency profile induce more variations on output signal when compared to GCN. The full frequency profile is not symmetric, because these convolution kernels are not symmetric. However, the variation on frequency profile might not be sufficient in problems that need some specific band-pass filters.

4. Conclusion

The spectral analysis showed that most influential graph convolutions GAT and GCN operate as low-pass filters. Interestingly, while being restricted to low-pass filters, they still obtain state-of-the-art performance on particular node classification problems such as Cora and Citeseer (Yang et al., 2016). These results on these particular problems are induced by the nature of the graphs to be processed. Indeed, citation network problems are inherently low-pass filtering problems, similarly to image segmentation problems, which are efficiently tackled by low-pass filtering. It is worth noting that, if we use enough convolution kernels, the frequency responses of ChebNet and CayleyNet cover nearly all frequency profiles. However, these frequency responses are not specific to special bands of frequency. It means that they can act as high-pass filters, but not as Gabor-like special band-pass filters.

Acknowledgments

This work was partially supported by the ANR grant APi (ANR-18-CE23-0014) and the PAUSE Program.

References

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4): 18–42, July 2017.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203, 2013.

Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C., and Murphy, K. Machine learning on graphs: A model and comprehensive taxonomy, 2020.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pp. 3844–3852, 2016.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyal, O., and Dahl, G. E. Neural message passing from quantum chemistry. In *Proceedings of the International Conference on Machine Learning*, 2017.

Graves, A., Mohamed, A.-r., and Hinton, G. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649. IEEE, 2013.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

Levie, R., Monti, F., Bresson, X., and Bronstein, M. M. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, Jan 2019. ISSN 1941-0476. doi: 10.1109/TSP.2018.2879624.

NT, H. and Maehara, T. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, December 2009.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Wu, F., Zhang, T., Souza Jr, A. H. d., Fifty, C., Yu, T., and Weinberger, K. Q. Simplifying graph convolutional networks. *arXiv preprint arXiv:1902.07153*, 2019a.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. A comprehensive survey on graph neural networks. arXiv preprint arXiv:1901.00596, 2019b.

Yang, Z., Cohen, W. W., and Salakhutdinov, R. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning, ICML'16*, 2016.

Supplementary Material

When Spectral Domain Meets Spatial Domain in Graph Neural Networks

Muhammet Balcilar, et al.

A. Theoretical Analysis of Chebyshev Kernels Frequency Profile

In this appendix, we provide the expressions of the full and standard frequency profiles of the Chebyshev convolution kernels.

Theorem A.1. *The frequency profile of the first Chebyshev convolution kernel for any undirected arbitrary graph defined by $C^{(1)} = I$ can be defined by*

$$F_1(\boldsymbol{\lambda}) = \mathbf{1}, \tag{15}$$

where $\mathbf{1}$ denotes the vector of ones of appropriate size.

Proof. When the identity matrix is used as convolution kernel, it just directly transmits the inputs to the outputs without any modification. This process is called all-pass filter. Mathematically, we can calculate the full frequency profile for kernel I by using Corollary 1.1, namely

$$F_1 = U^\top I U = U^\top U = I, \tag{16}$$

since the eigenvectors are orthonormal. Therefore, we can parameterize the diagonal of the full frequency profile by $\boldsymbol{\lambda}$ and reach the standard frequency profile as follows:

$$F_1(\boldsymbol{\lambda}) = \mathrm{diag}(I) = \mathbf{1}. \tag{17}$$

□

Theorem A.2. *The frequency profile of the second Chebyshev convolution kernel for any undirected arbitrary graph given by $C^{(2)} = 2L/\lambda_{\max} - I$ can be defined by*

$$F_2(\boldsymbol{\lambda}) = \frac{2\boldsymbol{\lambda}}{\lambda_{\max}} - \mathbf{1}. \tag{18}$$

Proof. We can compute the $C^{(2)}$ kernel full frequency profile using Corollary 1.1:

$$F_2 = U^\top \left(\frac{2}{\lambda_{\max}} L - I \right) U. \tag{19}$$

Since $U^\top I U = I$, (19) can be rearranged as

$$F_2 = \frac{2}{\lambda_{\max}} U^\top L U - I. \tag{20}$$

Since $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]$ are the eigenvalues of the graph Laplacian L, those must conform to the following condition:

$$LU = U\,\mathrm{diag}(\boldsymbol{\lambda}); \tag{21}$$

$$U^\top L U = \mathrm{diag}(\boldsymbol{\lambda}). \tag{22}$$

Replacing (22) into (20), we get

$$F_2 = \frac{2}{\lambda_{\max}} \mathrm{diag}(\boldsymbol{\lambda}) - I. \tag{23}$$

This full frequency profile consists of two parts, a diagonal matrix and the negative identity matrix. Therefore, we can parameterize the full frequency matrix diagonal to show the standard frequency profile as follows:

$$F_2(\boldsymbol{\lambda}) = \mathrm{diag}(F_2) = \frac{2\boldsymbol{\lambda}}{\lambda_{\max}} - \mathbf{1}. \tag{24}$$

□

Theorem A.3. *The frequency profile of third and followings Chebyshev convolution kernels for any undirected arbitrary graph can be defined by*

$$F_k = 2F_2F_{k-1} - F_{k-2}, \tag{25}$$

and their standard frequency profiles by

$$F_k(\boldsymbol{\lambda}) = 2F_2(\boldsymbol{\lambda})F_{k-1}(\boldsymbol{\lambda}) - F_{k-2}(\boldsymbol{\lambda}). \tag{26}$$

Proof. Given the third and following Chebyshev kernels defined by $C^{(k)} = 2C^{(2)}C^{(k-1)} - C^{(k-2)}$ and using Corollary 1.1, the corresponding frequency profile is

$$F_k = U^\top \left(2C^{(2)}C^{(k-1)} - C^{(k-2)} \right) U. \tag{27}$$

By expanding (27), we get

$$F_k = 2U^\top C^{(2)} C^{(k-1)} U - U^\top C^{(k-2)} U. \tag{28}$$

Since $UU^\top = I$, we can insert the product $UU^\top$ into (28). Thus, we have

$$F_k = 2U^\top C^{(2)} U U^\top C^{(k-1)} U - U^\top C^{(k-2)} U \tag{29}$$

$$F_k = 2\left(U^\top C^{(2)} U \right) \left(U^\top C^{(k-1)} U \right) - U^\top C^{(k-2)} U. \tag{30}$$

Since $F_{k'} = U^\top C^{(k')} U$ for any k', it follows that (30) and (25) are identical.

Hence F_1 and F_2 are diagonal matrices, and the rest of the kernels frequency profiles become diagonal matrices in (25). Therefore, we can write the corresponding standard frequency profiles of third and followings Chebyshev convolution kernels as follows:

$$F_k(\boldsymbol{\lambda}) = 2F_2(\boldsymbol{\lambda})F_{k-1}(\boldsymbol{\lambda}) - F_{k-2}(\boldsymbol{\lambda}). \tag{31}$$

□

B. Theoretical Analysis of CayleyNet Frequency Profile

CayleyNet uses the weight vector parametrization by $F_{i,j,l} = [g_{i,j,l}(\lambda_1, h), ..., g_{i,j,l}(\lambda_n, h)]^\top$, where the function $g(\cdot,\cdot)$ is defined in (Levie et al., 2019) by

$$g(\lambda, h) = c_0 + 2Re\left(\sum_{k=1}^{r} c_k \left(\frac{h\lambda - \mathbf{i}}{h\lambda + \mathbf{i}}\right)^k\right), \tag{32}$$

where $\mathbf{i}^2 = -1$, $Re(\cdot)$ is the function that returns the real part of a given complex number, c_0 is a trainable real coefficient, and $c_1, \ldots, c_r$ are complex trainable coefficients. We can write $h\lambda - \mathbf{i}$ in Euler form by $\sqrt{h^2\lambda^2+1}.e^{\mathbf{i}\,\mathrm{atan2}(-1,h\lambda)}$ and for $h\lambda + \mathbf{i}$ by $\sqrt{h^2\lambda^2+1}.e^{\mathbf{i}\,\mathrm{atan2}(1,h\lambda)}$. By this substitution, (32) becomes

$$g(\lambda, h) = c_0 + 2Re\left(\sum_{k=1}^{r} c_k e^{\mathbf{i}k(atan2(-1,h\lambda) - atan2(1,h\lambda))}\right). \tag{33}$$

where $atan2(y, x)$ is the inverse tangent function, which finds the angle (in range of $[-\pi, \pi]$) of a point given its y and x coordinates. For further simplification, let us introduce the $\theta(\cdot)$ function defined by

$$\theta(x) = atan2(-1, x) - atan2(1, x). \tag{34}$$

Since the c_ks are complex numbers, we can write them as a sum of real and imaginary parts, $c_k = a_k/2 + \mathbf{i}b_k/2$ (the scale factor 2 is added for convenience). Thus, (33) can be rewritten as follows:

$$g(\lambda, h) = c_0 + Re\left(\sum_{k=1}^{r} (a_k + \mathbf{i}b_k) e^{\mathbf{i}k\theta(h\lambda)}\right). \tag{35}$$

We can replace $e^{\mathbf{i}k\theta(h\lambda)}$ with its polar coordinate equivalence form $\cos(k\theta(h\lambda)) + \mathbf{i}\sin(k\theta(h\lambda))$. When we remove the imaginary components because of $Re(\cdot)$ function, (35) becomes

$$g(\lambda, h) = c_0 + \sum_{k=1}^{r} a_k \cos(k\theta(h\lambda)) - b_k \sin(k\theta(h\lambda)). \tag{36}$$

In this definition, there is no complex coefficient, but only real coefficients (c_0, a_k and b_k for $k = 1, \ldots, r$) to be tuned by training. By using the form in (36), we can parametrize CayleyNet by the parametrization matrix $B \in \mathbb{R}^{n \times 2r+1}$ by

$$[g(\lambda_0, h), \ldots, g(\lambda_n, h)]^\top = B[c_0, a_1, b_1, \ldots, a_r, b_r]^\top. \tag{37}$$

The s-th column vector of matrix B, denotes B_s, must fulfill the following conditions:

$$B_s = F_s(\boldsymbol{\lambda}) = \begin{cases} \mathbf{1} & \text{if } s = 1 \\ \cos(\frac{s}{2}\theta(h\boldsymbol{\lambda})) & \text{if } s \in \{2, 4, \ldots, 2r\} \\ -\sin(\frac{s-1}{2}\theta(h\boldsymbol{\lambda})) & \text{if } s \in \{3, 5, \ldots, 2r+1\} \end{cases} \tag{38}$$

We can see CayleyNet as a spectral graph convolution that uses $2r + 1$ convolution kernels. The first kernel is an all-pass filter, and the frequency profiles of remaining $2r$ kernels ($F_s(\boldsymbol{\lambda})$) are created using sine and cosine functions, with a parameter h used to scale the eigenvalues in (38). Considering (5) in Theorem 1, we can write CayleyNet's convolutions ($C^{(s)}$) in spatial domain. CayleyNet includes the tuning of this scaling parameter in the training pipeline. Note that because of the function definition in (34), $\theta(h\lambda)$ is not linear in λ. Therefore, F_s cannot be a perfect sinusoidal in λs.

C. Theoretical Analysis of GCN Frequency Profile

In this appendix, we study the GCN and its convolution kernel. We start by deriving the expression of its frequency profile.

Theorem C.1. *The frequency profile of GCN convolution kernel is defined by*

$$C_{GCN} = \widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}, \tag{39}$$

and can be written as

$$F_{GCN}(\boldsymbol{\lambda}) = \mathbf{1} - \frac{p}{p+1}\boldsymbol{\lambda}, \tag{40}$$

where $\boldsymbol{\lambda}$ is the eigenvalues of the normalized graph Laplacian and the given graph is an undirected regular graph whose node degrees are all equal to p.

Proof. Since $\widetilde{D}_{i,i} = \sum_j \widetilde{A}_{i,j}$ and $\widetilde{A} = (A + I)$, we can rewrite (39) as:

$$C_{GCN} = (D + I)^{-1/2}(A + I)(D + I)^{-1/2}. \tag{41}$$

Under the assumption that all node degrees are equal to p, we can write the diagonal degree matrix by $D = pI$. Then, (41) can be rewritten as

$$C_{GCN} = ((p + 1)I)^{-1/2}(A + I)((p + 1)I)^{-1/2}, \tag{42}$$

which is equivalent to

$$C_{GCN} = \frac{A+I}{p+1}. \tag{43}$$

Using Corollary 1.1, we can express the frequency profile of C_{GCN} in matrix form by

$$F_{GCN} = \frac{1}{p+1} U^\top A U + \frac{1}{p+1} I. \tag{44}$$

Since $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]$ are the eigenvalues of the normalized graph Laplacian $L = I - D^{-1/2} A D^{-1/2}$, they must conform to the following condition:

$$\left(I - D^{-1/2} A D^{-1/2}\right) U = U \operatorname{diag}(\boldsymbol{\lambda}). \tag{45}$$

According to $D = pI$, it conforms to $D^{-1/2} A D^{-1/2} = A/p$. Thus, (45) can be written as

$$U - \frac{AU}{p} = U \operatorname{diag}(\boldsymbol{\lambda}). \tag{46}$$

Then AU is expressed as

$$AU = pU - pU \operatorname{diag}(\boldsymbol{\lambda}) \tag{47}$$

Replacing AU in (44), we obtain

$$F_{GCN} = \frac{1}{p+1} U^\top \left(pU - pU \operatorname{diag}(\boldsymbol{\lambda})\right) + \frac{1}{p+1} I. \tag{48}$$

Since $U^\top U = I$, then we have

$$F_{GCN} = \frac{pI - p \operatorname{diag}(\boldsymbol{\lambda}) + I}{p+1}. \tag{49}$$

This expression can be simplified to

$$F_{GCN} = I - \frac{p}{p+1} \operatorname{diag}(\boldsymbol{\lambda}), \tag{50}$$

which is equal to the matrix form defined in (40) since $F_{GCN}(\boldsymbol{\lambda}) = \operatorname{diag}(F_{GCN})$. □

This demonstration shows that the GCN frequency profile acts as a low-pass filter. When the given graph is a circular undirected graph, all node degrees are equal to $p = 2$, leading to a frequency profile defined by $\mathbf{1} - 2\boldsymbol{\lambda}/3$. Since the normalized graph Laplacian eigenvalues are in the range $[0, 2]$, the filter magnitude linearly decreases until the third quarter of the spectrum (cut-off frequency) where it reaches zero. Then it linearly increases until the end of the spectrum. This explains the shape of the frequency profile of GCN convolutions for 1D regular graph observed in Figure 3.

However, this conclusion cannot explain the perturbations on the GCN frequency profile. To analyse this point, we relax the assumption $D = pI$ and rewrite (41) as

$$C_{GCN} = (D+I)^{-1} + (D+I)^{-1/2} A (D+I)^{-1/2}. \tag{51}$$

We can see that the GCN kernel consists of two parts, $C_{GCN} = c_1 + c_2$, where first part is given by $c_1 = (D + I)^{-1}$ and the second one is $c_2 = (D+I)^{-1/2} A (D+I)^{-1/2}$.

For the second part (c_2), we can write it using the element-wise multiplication operator $\odot$ (Hadamard multiplication)

$$c_2 = A \odot \sqrt{\mathbf{1}/(d+1)} \cdot \sqrt{\mathbf{1}/(d+1)}^\top, \tag{52}$$

where d is the column degree vector $d = \operatorname{diag}(D)$ and the division and square-root are also element-wise (Hadamard) operations. With the same notation, we can rewrite the Chebyshev second kernel, assuming that $\lambda_{\max} = 2$,

$$C^{(2)} = -A \odot \sqrt{\mathbf{1}/d} \cdot \sqrt{\mathbf{1}/d}^\top. \tag{53}$$

The two expressions (52) and (53) show that negative c_2 is an approximation of the second Chebyshev kernel if vector d consists of same values, as it was assumed in Theorem C.1. When the vector d is composed of different values, the two matrices $\sqrt{\mathbf{1}/d}.\sqrt{\mathbf{1}/d}^\top$ and $\sqrt{\mathbf{1}/(d+1)}.\sqrt{\mathbf{1}/(d+1)}^\top$ are not proportional for each coordinate (i.e., entry). To obtain c_2 from $C^{(2)}$, we need to use different coefficients for each coordinate of the kernel. If the difference between node degrees is important, these coefficients have the strong influence, and c_2 may be very different from $C^{(2)}$. Conversely, if the node degrees are quite uniform, these coefficients may be neglected. This phenomenon is the first cause of perturbation on GCN frequency profile.

The first part (c_1) of the GCN kernel in (51) is more interesting. Actually, it is a diagonal matrix that shows the contribution of each node in the convolution process. Instead of looking for some approximations of known frequency profiles such as those of Chebyshev kernels, we can write its frequency profile directly. Using Corollary 1.1, we can express the frequency profile of c_1 in matrix form by

$$F_{c_1} = (U^\top c_1 U), \tag{54}$$

where U is the eigenvectors matrix. By taking advantage of having a diagonal kernel c_1, we can express each component of full frequency profile as

$$F_{c_1}(i,j) = \sum_{k=1}^{n} \left(\frac{1}{1+d_k} U_{i,k} U_{j,k} \right), \tag{55}$$

where n is the number of nodes in the graph, d_k is degree of the k-th node, $U_{i,k}$ is the k-th element of i-th eigenvector. As eigenvectors U_i and U_j are orthogonal for $i \neq j$, their scalar product is null. However, in (55), the weighting coefficient $\frac{1}{1+d_k}$ is not constant over all the dimensions of the eigenvectors. Therefore, there is no guarantee that $F_{c_1}(i,j)$ is null. This is another reason that explains that the GCN frequency profile has many non-zero elements outside of the diagonal.

In addition, it is also clear that the standard frequency profile of c_1 (diagonal of F_{c_1}, i.e., $F_{c_1}(i,i)$ in (55)) is not smooth. Indeed, the diagonal elements of F_{c_1} can be written as a weighted sum of squared eigenvalues elements, which again is weighted by $1/(1+d_k)$. If the latter is constant for all k, the sum of squared eigenvectors elements has to be 1 since the eigenvectors have unit L2-norm. But in the general case where $1/(1+d_k)$ are not necessarily constant over all the dimensions of eigenvectors, the diagonal of the matrix may have some perturbations. This point constitutes another explanation on the fact that the GCN standard frequency profile is not smooth.

On the other hand, under the assumption that the node degrees distribution is uniform, we can derive the following approximation:

$$p \approx \overline{d} = \frac{1}{n}\sum_{k=1}^{n} d_k. \tag{56}$$

We can then write an approximation of the GCN frequency profile as a function of the average node degree by replacing p with $\overline{d}$ in (47) and obtain the final approximation:

$$F_{GCN}(\boldsymbol{\lambda}) \approx \mathbf{1} - \frac{\overline{d}}{\overline{d}+1}\boldsymbol{\lambda}. \tag{57}$$

We can theoretically show the cut-off frequency where GCN kernel's frequency profile reach 0 by

$$\lambda_{\text{cut}} \approx \frac{\overline{d}+1}{\overline{d}}. \tag{58}$$