# Analyzing the Expressive Power of Graph Neural Networks in a Spectral Perspective

Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, Paul Honeine

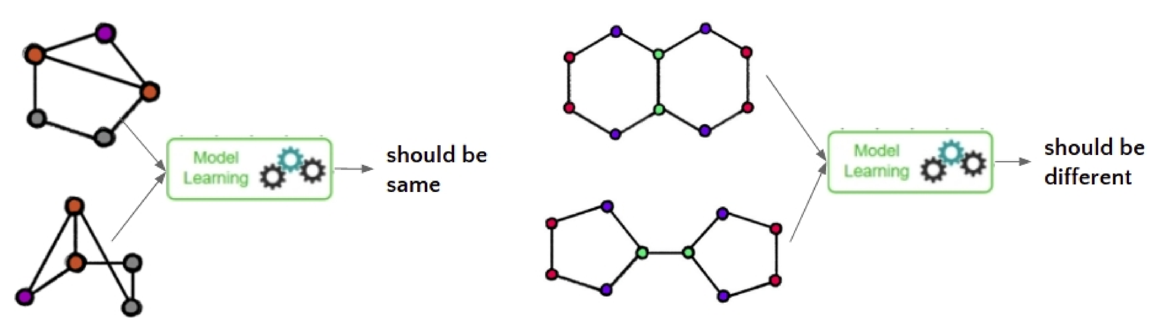
Normandy University, LITIS Lab, University of Rouen Normandy, INSA Rouen Normandie, Rouen, 76000, France

## Contributions

- Bridging the gap between spectral and spatial domains in GNN by demonstrating the equivalence of graph convolution processes.
- Propose a new general framework and taxonomy for GNNs.
- Provide theoretical and empirical spectral analysis of GNN models.

## Introduction

- Universality of the GNN depends on permutational invariance and ability to produce different outputs for non-isomorphic graphs.




- Expressive power of GNNs were evaluated using equivalence of Weisfeiler-Lehman test order so far.
- MPNN is known to be 1-WL equivalent and WL test order does not tell any differences between MPNNs.
- Since GNN can be seen as a signal processing pipeline, analyzing GNN models in a spectral point of view can bring a new perspective on their expressive power.

## Spectral and Spatial GNNs

- **Spectral GNN:** Relies on eigendecomposition of graph Laplacian:

$$H_j^{(l+1)} = \sigma\left(\sum_{i=1}^{f_l} U \mathrm{diag}(F_i^{(l,j)}) U^\top H_i^{(l)}\right)$$

- To overcome drawbacks, trainable weights are parametrized by $F_i^{(l,j)} = B\left[W_{i,j}^{(l,1)}, \ldots, W_{i,j}^{(l,s_e)}\right]^\top$
- **Spatial GNN:** Consider an *agg* operator, which aggregates the neighborhood nodes, and an *upd* operator updates the concerned node as follows:

$$H_{:v}^{(l+1)} = upd\left(g_0(H_{:v}^{(l)}), agg\left(g_1(H_{:u}^{(l)}) : u \in \mathcal{N}(v)\right)\right)$$

## Bridging Spatial and Spectral Domains

- $upd(x, y) = \sigma(x + y)$, *agg* is a sum of neighbor nodes and $g_i$ applies a linear transformation, Spatial GNN can be written by our general framework by:

$$H^{(l+1)} = \sigma\left(\sum_s C^{(s)} H^{(l)} W^{(l,s)}\right)$$

- **Theorem** Spectral GNN parameterized with $B$ of entries $B_{i,j} = \Phi_j(\lambda_i)$ is a particular case of our general framework with the convolution kernel $C^{(s)}$:

$$C^{(s)} = U(\Phi_s(\lambda))U^\top.$$

- **Corollary** The frequency profile of a graph convolution support can be defined in spectral domain by:

$$\Phi_s(\lambda) = \mathrm{diag}^{-1}(U^\top C^{(s)} U).$$

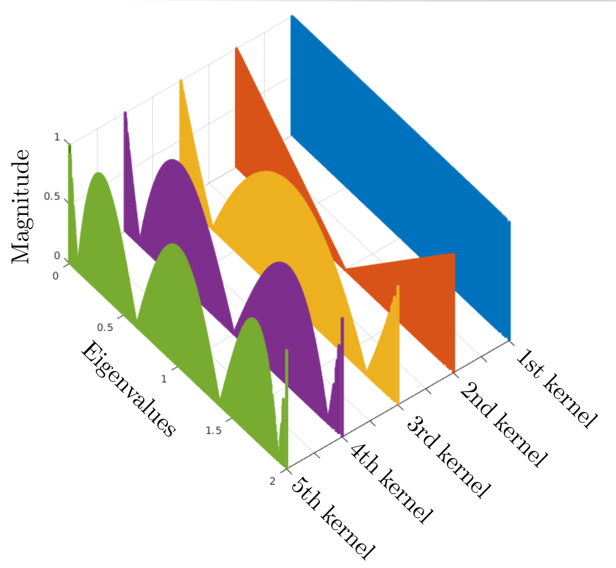
## New Taxonomy of GNN

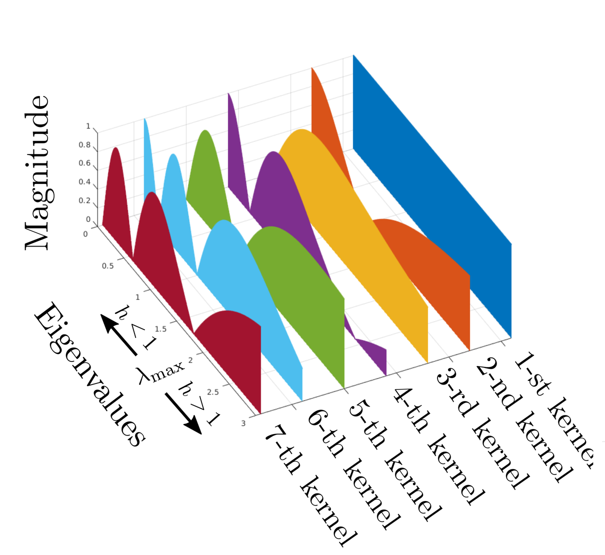
- **Definition 1:** A **Trainable-support** is a Graph Convolution Support $C^{(s)}$ with at least one trainable parameter that can be tuned during training. If $C^{(s)}$ has no trainable parameters, *i.e.* when the supports are pre-designed, it is called a ***fixed-support*** graph convolution.
- **Definition 2: Spectral-designed** graph convolution refers to a convolution where supports are written as a function of eigenvalues ($\Phi_s(\lambda)$) and eigenvectors ($U$) of the corresponding graph Laplacian. Thus, each convolution support $C^{(s)}$ has the same frequency response $\Phi_s(\lambda)$ over different graphs. Convolution out of this definition is called **spatial-designed** graph convolution.

## Classification of GNN models and their theoretical frequency responses.

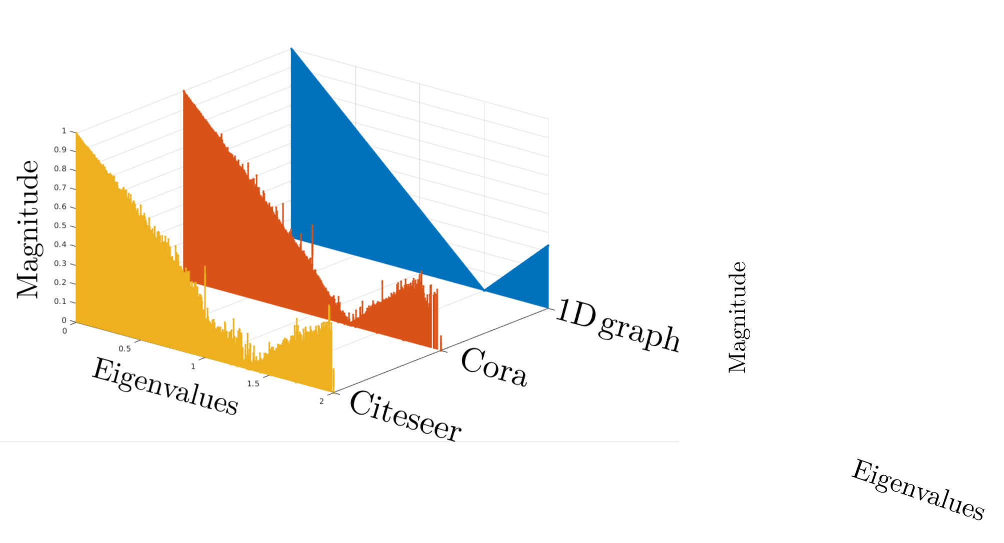
| | Design | Support Type | Convolution Matrix | Frequency Response |
|---|---|---|---|---|
| MLP | Spectral | Fixed | $C = I$ | $\Phi(\lambda) = 1$ |
| GCN | Spatial | Fixed | $C = \tilde{D}^{-0.5}\tilde{A}\tilde{D}^{-0.5}$ | $\Phi(\lambda) \approx 1 - \lambda p/(p+1)$ |
| GIN | Spatial | Trainable | $C = A + (1+\epsilon)I$ | $\Phi(\lambda) \approx p\left(\frac{1+\epsilon}{p} + 1 - \lambda\right)$ |
| GAT | Spatial | Trainable | $C^{(s)}_{v,u} = e_{v,u} / \sum_{k \in \tilde{N}(v)} e_{v,k}$ | NA |
| CayleyNet | Spectral | Trainable | $C^{(1)} = I$<br>$C^{(2r)} = Re(\rho(hL)^r)$<br>$C^{(2r+1)} = Re(i\rho(hL)^r)$ | $\Phi_1(\lambda) = 1$<br>$\Phi_{2r}(\lambda) = \cos(r\theta(h\lambda))$<br>$\Phi_{2r+1}(\lambda) = -\sin(r\theta(h\lambda))$ |
| ChebNet | Spectral | Fixed | $C^{(1)} = I$<br>$C^{(2)} = 2L/\lambda_{\max} - I$<br>$C^{(s)} = 2C^{(2)}C^{(s-1)} - C^{(s-2)}$ | $\Phi_1(\lambda) = 1$<br>$\Phi_2(\lambda) = 2\lambda/\lambda_{\max} - 1$<br>$\Phi_s(\lambda) = 2\Phi_2(\lambda)\Phi_{s-1}(\lambda) - \Phi_{s-2}(\lambda)$ |

## ChebNet and CayleyNet Freq. Res.

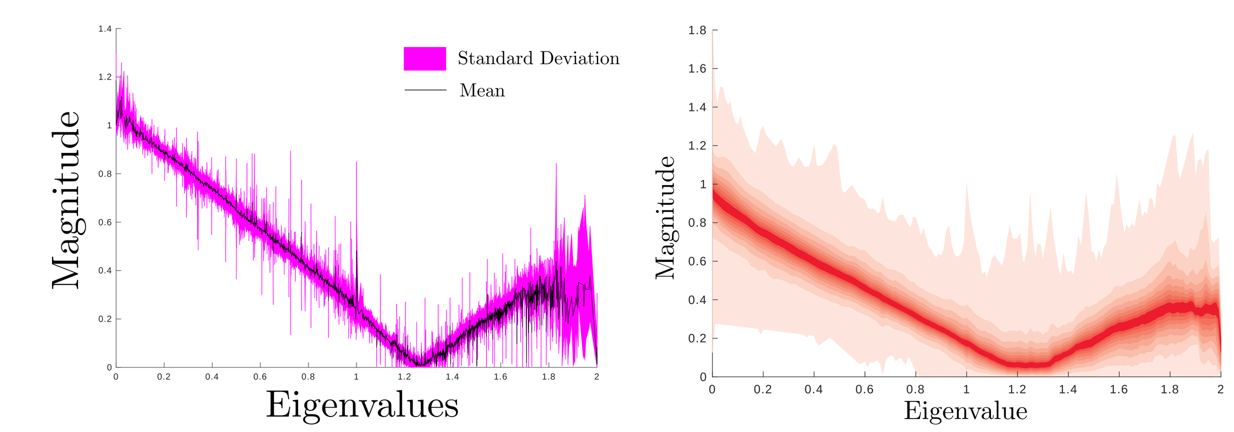



## GCN and GIN Freq. Res.




## GAT Simulated Freq. Res.



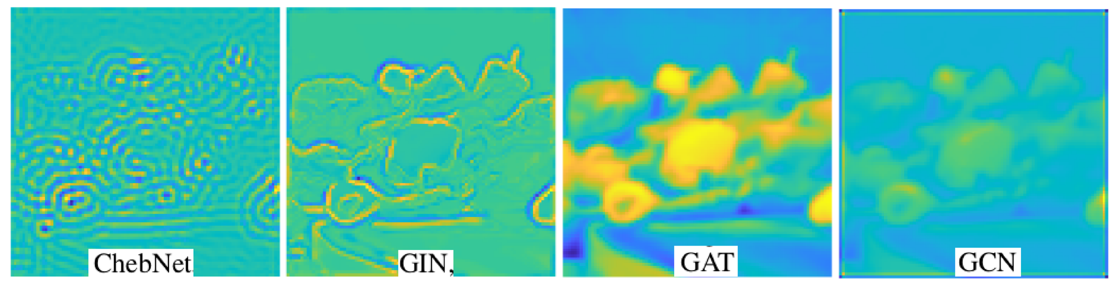

## Conclusion of Spectral Analysis

- Spatial designs are nothing but just low-pass filters!
- Spectral designs cover the spectrum well but they do not have band specific filters.

## Experimental Results

- **2DGrid Graph** is a node regression problem for low-pass, band-pass and high-pass filtered image. Results are in sum of squared error.

| Prediction Target | GCN | GIN | GAT | ChebNet |
|---|---|---|---|---|
| Low-pass filter ($\Phi_1$) | 15.55 | 11.01 | 10.50 | 3.44 |
| Band-pass filter ($\Phi_2$) | 79.72 | 63.24 | 79.68 | 17.30 |
| High-pass filter ($\Phi_3$) | 29.51 | 14.27 | 29.10 | 2.04 |




- **BandPass Graph** is a binary classification problem according to frequency that the graph signal carries.

| | MLP | GCN | GIN | GAT | ChebNet |
|---|---|---|---|---|---|
| Accuracy | 50 | 77.90 | 87.60 | 85.30 | 98.2 |
| Loss | 0.69 | 0.454 | 0.273 | 0.324 | 0.062 |

- **Mnist-75** is a graph classification problem. It is a superpixel graph version of MNIST dataset.

| | MLP | GCN | GIN | GAT | CayleyNet | ChebNet |
|---|---|---|---|---|---|---|
| Accuracy | 25.10 | 52.98 | 75.23 | 82.73 | 90.31 | 92.08 |

## Acknowledgments

This work was partially supported by the French Agence National de Recherche (ANR), grant APi (ANR-18-CE23-0014), and the PAUSE Program (Collège de France).